Generic Construction of Broadcast Authenticated Encryption with Keyword Search*

Keita Emura[§]

[§]Kanazawa University, Japan.[†]

November 9, 2023

Abstract

As a multi-receiver variant of public key authenticated encryption with keyword search (PAEKS), broadcast authenticated encryption with keyword search (BAEKS) was proposed by Liu et al. (ACISP 2021). BAEKS focuses on receiver anonymity, where no information about the receiver is leaked from ciphertexts, which is reminiscent of the anonymous broadcast encryption. Here, there are rooms for improving their security definitions, e.g., two challenge sets of receivers are selected before the setup phase, and an adversary is not allowed to corrupt any receiver. In this paper, we propose a generic construction of BAEKS derived from PAEKS that provides ciphertext anonymity and consistency in a multi-receiver setting. The proposed construction is an extension of the generic construction proposed by Libert et al. (PKC 2012) for the anonymous broadcast encryption and provides adaptive corruptions. We also demonstrate that the Qin et al. PAEKS scheme (ProvSec 2021) provides ciphertext anonymity and consistency in a multi-receiver setting and can be employed as a building block of the proposed generic construction. Moreover, we demonstrate that the Mukherjee BAEKS scheme (ACISP 2023) can be employed as a building block of the proposed generic construction.

1 Introduction

Public key authenticated encryption with keyword search (PAEKS) [8, 12–15, 20, 28, 32–35, 37] has been proposed as an extension of public key encryption with keyword search (PEKS) [7]. In PAEKS, a sender secret key is required for encryption. Because of the restriction of the rights of encryption, a keyword guessing attack[1] is prevented. PAEKS requires that no information about the keyword is leaked from both the ciphertexts and trapdoors.

*The journal version appeared at IET Information Security with the title "Generic Construction of Fully Anonymous Broadcast Authenticated Encryption with Keyword Search with Adaptive Corruptions" [16]. In this ePrint version, we add the Mukherjee BAEKS scheme [30] as an independent and concurrent work. Moreover, we demonstrate that the Mukherjee PAEKS scheme (i.e., the Mukherjee BAEKS scheme with the single receiver) can be employed as a building block of the proposed generic construction.

[†]The main part of study was done when the author was with the National Institute of Information and Communications Technology (NICT), Japan.

[1]In PEKS, if an adversary has a trapdoor, information about which keyword is associated with the trapdoor is leaked by running a test algorithm with self-made ciphertexts. This keyword guessing attack is unavoidable in PEKS because anyone can generate a ciphertext of any keyword, and anyone can run the test algorithm when they obtain the trapdoor.

Broadcast authenticated encryption with keyword search (BAEKS) [27] was proposed by Liu et al. as a multi-receiver variant of PAEKS. Unlike other multi-receiver variants of P(A)EKS [3,5,9,19, 21,22,29,36,38],[2] BAEKS focuses on receiver anonymity, where no information about the receiver is leaked from ciphertexts, which is reminiscent of the anonymous broadcast encryption [6,18,23–26]. BAEKS also considers trapdoor anonymity. The flow of BAEKS is described below. A sender generates a ciphertext by specifying a set of receivers S and a keyword to be encrypted kw, and sends the ciphertext to a cloud server. Each receiver generates a trapdoor by specifying a sender and a keyword to be searched kw', and sends the trapdoor to the cloud server. The cloud server runs a test algorithm, and forwards the corresponding content[3] to a receiver based on the result of the test algorithm. Informally, the BAEKS scheme is correct if the test algorithm outputs 1 when $kw = kw'$ and the trapdoor is generated by a receiver belonging to S.

Liu et al. proposed a pairing-based BAEKS scheme (in the random oracle model). However, the following restrictions in their security definitions can be observed:

1. No consistency is defined, i.e., it is not formally defined when the test algorithm outputs 0.
   - If a PAEKS scheme needs to provides correctness only, a meaningless scheme can be constructed as follows. The encryption and trapdoor generation algorithms output random values, and the test algorithm always outputs 1 regardless of the input. Then, no information about the keyword is revealed from both the ciphertext and trapdoor, and the construction provides correctness. To avoid this meaningless construction, consistency is important in the searchable encryption context.
   - We note that Liu et al. construction defines when the test algorithm outputs 0. Also, we do not claim that their scheme does not provide consistency.
2. The challenge sets S_0^* and S_1^* are fixed during the setup phase. Furthermore, the two challenge sets contain only one distinct receiver public key and other identical receiver public keys. This restricts the attack strategies of adversaries.
3. An adversary is not allowed to obtain the secret key of a receiver, i.e., no corruption is allowed.

As an independent and concurrent work, Mukherjee [30] proposed a pairing-based BAEKS scheme which is secure in the standard model. Security models are improved, especially consistency in the multi-sender setting defined in [15] is extended to BAEKS (as in our definition) and statistical consistency is considered (which is stronger than our computational consistency). However, there is room for improvement in terms of anonymity, where an adversary is not allowed to obtain the secret key of a receiver, i.e., no corruption is allowed.

To this day, no generic BAEKS construction has been proposed. Since generic constructions of anonymous broadcast encryption have been proposed, it is reasonable to consider whether generic constructions of anonymous broadcast encryption can be customized for BAEKS or not.

[2] Attrapadung et al. [5] introduced broadcast encryption with keyword search (BEKS) whose security is defined as a selective manner. Chatterjee and Mukherjee [9] proposed a BEKS scheme which is secure under the SXDH (Symmetric eXternal Diffie-Hellman) assumption and provides adaptive security. They also mentioned that the generic construction of Ambrona et al. [4] on [10] or on [11] also provide pairing-based BEKS constructions. Note that Chatterjee and Mukherjee called a BEKS scheme anonymous, if the challenge ciphertext hides associated challenge keyword. Moreover, in the BEKS syntax, the test algorithm takes a set of receivers. Thus these BEKS constructions do not provide receiver anonymity.

[3] In a real system, additional encryption is required to encrypt a content. For example, a content is encrypted by an anonymous broadcast encryption scheme, and keywords are encrypted by a searchable encryption scheme. Then, the cloud server sends a ciphertext of the content to a receiver based on the result of the test algorithm. As in Liu et al.'s paper [27], we only focus on the searching phase in this paper.

Table 1: Comparison between our instantiations from the Liu et al. BAEKS scheme [27], and the Mukherjee BAEKS scheme [30]. Let S be a set of receivers specified in the encryption algorithm and $N = |S|$. CT and TD stand for ciphertext and trapdoor, respectively. ROM and STD stand for random oracle model and standard model, respectively. We emphasize that our generic construction provides trapdoor anonymity if the underlying PAEKS scheme provides trapdoor anonymity. Though an adversary is allowed to obtain the trapdoors of the receivers belonging to the challenge set in [30], no corruption is allowed. Thus, we state it as Restricted* in the table.

Scheme	CT Size	Test Attempts	Consistency	CT Anon.	TD Anon.	Corruption	ROM/ STD
Liu et al. [27]	$O(N)$	$O(N)$	Not Defined	Restricted	Yes	No	ROM
Mukherjee [30]	$O(N)$	$O(N)$	Defined	Restricted*	Yes	No	STD
Ours (§ 4) + [35]	$O(N)$	$O(N)$	Defined	Full	No	Adaptive	ROM
Ours (§ 4) + [30]	$O(N)$	$O(N)$	Defined	Full	Yes	Adaptive	STD

Anonymous Broadcast Encryption. Here, we revisit a generic construction of anonymous broadcast encryption to investigate the properties required to construct BAEKS by extending this generic construction.[4] Libert et al. [26] proposed a generic construction (under adaptive corruptions) that provides full anonymity, where no information about the receiver is leaked from ciphertexts, even against ciphertext receivers; i.e., an adversary is allowed to obtain the secret keys of the receivers belonging to $S_0^* \cap S_1^*$ where S_0^* and S_1^* are the challenge sets. Specifically, an adversary is not allowed to obtain the secret keys of the receivers belonging to S where $S \cap (S_0^* \triangle S_1^*) = \emptyset$ (here, $S_0^* \triangle S_1^*$ is the symmetric difference defined as $S_0^* \triangle S_1^* = (S_0^* \setminus S_1^*) \cup (S_1^* \setminus S_0^*)$). The construction assumes that the underlying encryption scheme is key private, i.e., the public key used for encryption is not leaked from ciphertexts. Furthermore, the underlying encryption scheme is required to be (weakly) robust [1, 2], i.e., the decryption algorithm outputs the error symbol $\perp$ when a non-appropriate decryption key is used for decryption. Specifically, for two distinct key pairs (pk, sk) and (pk', sk'), the decryption result of a ciphertext generated by pk is $\perp$ when sk' is used for decryption. Robustness is important in identifying which ciphertext can be decrypted by receivers because of the key privacy. At a high level, a ciphertext is a set of ciphertexts of the underlying encryption scheme (with random permutations of ciphertexts). When a receiver decrypts a ciphertext, the receiver decrypts each ciphertext of the underlying encryption scheme one by one and outputs a non-$\perp$ decryption result.

Towards Generic Construction of BAEKS. Intuitively, BAEKS can be genetically constructed from PAEKS if the underlying PAEKS scheme provides anonymity. In addition to anonymity, we should pay attention to the robustness in the PAEKS context. That is, we need to ensure that a trapdoor generated by a receiver secret key should not work against ciphertexts generated by the public key of another receiver, even if the same keyword is associated. However, previous PAEKS schemes only considered the following case: $kw \neq kw'$ where kw is used to generate a ciphertext and kw' is used to generate a trapdoor. One exception is consistency in the multi-sender setting defined in [15] where a trapdoor associated with a sender does not work against ciphertexts generated by the secret key of another sender, even if the same keyword is associated. Thus, we need to consider the dual concept, i.e., consistency in the multi-receiver setting.

Our Contribution. In this paper, we propose a generic construction of BAEKS derived from PAEKS that provides ciphertext and trapdoor anonymity as well as consistency in a multi-receiver

[4] We do not consider chosen-ciphertext attack (CCA) in this paper and we omit one-time signatures from the construction hereafter.

setting. The proposed construction is an extension of the generic construction of the anonymous broadcast encryption [26] and provides adaptive corruptions.

We also demonstrate that the Qin et al. PAEKS scheme [35] provides consistency in a multi-receiver setting and ciphertext anonymity. A comparison of our instantiations with the Liu et al. BAEKS scheme [27] and the Mukherjee BAEKS scheme [30] is presented in Table 1. We note that BAEKS implies PAEKS and the Mukherjee BAEKS scheme can be employed as a building block of the proposed generic construction since the Mukherjee BAEKS scheme provides consistency in a multi-receiver setting, as mentioned in [30]. The number of test attempts and the ciphertext size are the same as those reported by Liu et al. and Mukherjee, although the proposed construction provides a higher security level in terms of ciphertext anonymity and adaptive corruptions.

We note that the Qin et al. PAEKS scheme does not provide trapdoor anonymity. Consequently, our instantiation based on the Qin et al. PAEKS scheme does not provide trapdoor anonymity. However, we argue that trapdoor anonymity is not necessary, at least for the setting considered in [27]. We recall that a cloud server forwards the corresponding content to a receiver based on the result of the test algorithm. Then, the cloud server needs to know the destination, i.e., it needs to obtain information about the receivers. If not, there is no way to send the content to the receivers. Although we do not deny the possibility that some applications may require trapdoor anonymity, we do not consider trapdoor anonymity in our instantiation. We emphasize that the proposed generic construction provides trapdoor anonymity if the underlying PAEKS scheme provides trapdoor anonymity. Thus, our instantiation from the Mukherjee BAEKS scheme provides trapdoor anonymity.

2 Definitions of PAEKS in the Multi-Receiver Setting

In this section, we introduce the definitions of PAEKS in the multi-receiver setting.

Definition 1 (Syntax of PAEKS)**.** *A PAEKS scheme* PAEKS *consists of the following six algorithms* $(\mathsf{PAEKS.Setup}, \mathsf{PAEKS.KG_R}, \mathsf{PAEKS.KG_S}, \mathsf{PAEKS.Enc}, \mathsf{PAEKS.Trapdoor}, \mathsf{PAEKS.Test})$ *defined as follows.*

$\mathsf{PAEKS.Setup}$**:** *The setup algorithm takes a security parameter* λ *as input, and outputs a common parameter* pp*. We assume that* pp *implicitly contains the keyword space* $\mathcal{KS}$*.*

$\mathsf{PAEKS.KG_R}$**:** *The receiver key generation algorithm takes* pp *as input, and outputs a public key* $\mathsf{pk_R}$ *and secret key* $\mathsf{sk_R}$*.*

$\mathsf{PAEKS.KG_S}$**:** *The sender key generation algorithm takes* pp *as input, and outputs a public key* $\mathsf{pk_S}$ *and secret key* $\mathsf{sk_S}$*.*

$\mathsf{PAEKS.Enc}$**:** *The keyword encryption algorithm takes* $\mathsf{pk_R}$*,* $\mathsf{pk_S}$*,* $\mathsf{sk_S}$*, and a keyword* $kw \in \mathcal{KS}$ *as input, and outputs a ciphertext* $\mathsf{ct_{PAEKS}}$*.*

$\mathsf{PAEKS.Trapdoor}$**:** *The trapdoor algorithm takes* $\mathsf{pk_R}$*,* $\mathsf{pk_S}$*,* $\mathsf{sk_R}$*, and a keyword* $kw' \in \mathcal{KS}$ *as input, and outputs a trapdoor* $\mathsf{td}_{\mathsf{R},kw'}$*.*

$\mathsf{PAEKS.Test}$**:** *The test algorithm takes* $\mathsf{ct_{PAEKS}}$ *and* $\mathsf{td}_{\mathsf{R},kw'}$ *as input, and outputs 1 or 0.*

Definition 2 (Correctness)**.** *For any security parameter* λ*, any common parameter* $\mathsf{pp} \leftarrow \mathsf{PAEKS.Setup}(1^\lambda)$*, any key pairs* $(\mathsf{pk_R}, \mathsf{sk_R}) \leftarrow \mathsf{PAEKS.KG_R}(\mathsf{pp})$ *and* $(\mathsf{pk_S}, \mathsf{sk_S}) \leftarrow \mathsf{PAEKS.KG_S}(\mathsf{pp})$*, and any keyword* $kw \in \mathcal{KS}$*, let* $\mathsf{ct_{PAEKS}} \leftarrow \mathsf{PAEKS.Enc}(\mathsf{pk_R}, \mathsf{pk_S}, \mathsf{sk_S}, kw)$ *and* $\mathsf{td}_{\mathsf{R},kw} \leftarrow \mathsf{PAEKS.Trapdoor}(\mathsf{pk_R}, \mathsf{pk_S}, \mathsf{sk_R}, kw)$*. Then* $\Pr[\mathsf{PAEKS.Test}(\mathsf{ct_{PAEKS}}, \mathsf{td}_{\mathsf{R},kw}) = 1] = 1 - \mathsf{negl}(\lambda)$ *holds.*

Next, we define computational consistency in the multi-receiver setting which guarantees that a trapdoor generated by a receiver secret key does not work against ciphertexts generated by the public key of another receiver, even if the same keyword is associated. As in [15], the following definition can be extended to consider the multi-sender setting if necessary.

Definition 3 (Computational Consistency for Multi Receivers)**.** *For all probabilistic polynomial-time (PPT) adversaries $\mathcal{A}$, we define the following experiment.*

$$
\begin{aligned}
&\mathsf{Exp}^{\mathsf{consist}}_{\mathsf{PAEKS},\mathcal{A}}(\lambda):\\
&\quad \mathsf{pp} \leftarrow \mathsf{PAEKS.Setup}(1^\lambda)\\
&\quad (\mathsf{pk}_{\mathsf{R}[0]}, \mathsf{sk}_{\mathsf{R}[0]}) \leftarrow \mathsf{PAEKS.KG}_{\mathsf{R}}(\mathsf{pp});\ (\mathsf{pk}_{\mathsf{R}[1]}, \mathsf{sk}_{\mathsf{R}[1]}) \leftarrow \mathsf{PAEKS.KG}_{\mathsf{R}}(\mathsf{pp})\\
&\quad (\mathsf{pk}_{\mathsf{S}}, \mathsf{sk}_{\mathsf{S}}) \leftarrow \mathsf{PAEKS.KG}_{\mathsf{S}}(\mathsf{pp})\\
&\quad (kw, kw', i, j) \leftarrow \mathcal{A}(\mathsf{pp}, \mathsf{pk}_{\mathsf{R}[0]}, \mathsf{pk}_{\mathsf{R}[1]}, \mathsf{pk}_{\mathsf{S}})\ \ s.t.\ kw, kw' \in \mathcal{KS} \wedge i, j \in \{0,1\} \wedge (kw, i) \neq (kw', j)\\
&\quad \mathsf{ct}_{\mathsf{PAEKS}} \leftarrow \mathsf{PAEKS.Enc}(\mathsf{pk}_{\mathsf{R}[i]}, \mathsf{pk}_{\mathsf{S}}, \mathsf{sk}_{\mathsf{S}}, kw)\\
&\quad \mathsf{td}_{\mathsf{R}[j],kw'} \leftarrow \mathsf{PAEKS.Trapdoor}(\mathsf{pk}_{\mathsf{R}[j]}, \mathsf{pk}_{\mathsf{S}}, \mathsf{sk}_{\mathsf{R}[j]}, kw')\\
&\quad \textit{If } \mathsf{PAEKS.Test}(\mathsf{ct}_{\mathsf{PAEKS}}, \mathsf{td}_{\mathsf{R}[j],kw'}) = 1,\ \textit{then output } 1,\ \textit{and } 0\ \textit{otherwise.}
\end{aligned}
$$

We say that a PAEKS scheme PAEKS *is consistent if the advantage*

$$\mathsf{Adv}^{\mathsf{consist}}_{\mathsf{PAEKS},\mathcal{A}}(\lambda) := \Pr[\mathsf{Exp}^{\mathsf{consist}}_{\mathsf{PAEKS},\mathcal{A}}(\lambda) = 1]$$

is negligible in the security parameter λ.

Next, we define indistinguishability against the chosen keyword attack (IND-CKA) which ensures that no information about the keyword is leaked from ciphertexts. We also capture ciphertext anonymity simultaneously, i.e., $\mathsf{pk}^*_{\mathsf{R}[b]}$ is used for generating the challenge ciphertext. If we explicitly mention the IND-CKA security in the non-anonymous setting, then $(\mathsf{pk}^*_{\mathsf{R}[0]}, \mathsf{sk}^*_{\mathsf{R}[0]}) = (\mathsf{pk}^*_{\mathsf{R}[1]}, \mathsf{sk}^*_{\mathsf{R}[1]})$ in the following experiment.

Definition 4 (IND-CKA)**.** *For all PPT adversaries $\mathcal{A}$, we define the following experiment.*

$$
\begin{aligned}
&\mathsf{Exp}^{\mathsf{IND\text{-}CKA}}_{\mathsf{PAEKS},\mathcal{A}}(\lambda):\\
&\quad \mathsf{pp} \leftarrow \mathsf{PAEKS.Setup}(1^\lambda)\\
&\quad (\mathsf{pk}^*_{\mathsf{R}[0]}, \mathsf{sk}^*_{\mathsf{R}[0]}) \leftarrow \mathsf{PAEKS.KG}_{\mathsf{R}}(\mathsf{pp});\ (\mathsf{pk}^*_{\mathsf{R}[1]}, \mathsf{sk}^*_{\mathsf{R}[1]}) \leftarrow \mathsf{PAEKS.KG}_{\mathsf{R}}(\mathsf{pp})\\
&\quad (\mathsf{pk}_{\mathsf{S}}, \mathsf{sk}_{\mathsf{S}}) \leftarrow \mathsf{PAEKS.KG}_{\mathsf{S}}(\mathsf{pp})\\
&\quad (kw^*_0, kw^*_1, \mathsf{state}) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{pp}, \mathsf{pk}^*_{\mathsf{R}[0]}, \mathsf{pk}^*_{\mathsf{R}[1]}, \mathsf{pk}_{\mathsf{S}})\ \ s.t.\ kw^*_0, kw^*_1 \in \mathcal{KS} \wedge\ kw^*_0 \neq kw^*_1\\
&\quad b \xleftarrow{\$} \{0,1\};\ \mathsf{ct}^*_{\mathsf{PAEKS}} \leftarrow \mathsf{PAEKS.Enc}(\mathsf{pk}^*_{\mathsf{R}[b]}, \mathsf{pk}_{\mathsf{S}}, \mathsf{sk}_{\mathsf{S}}, kw^*_b)\\
&\quad b' \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{state}, \mathsf{ct}^*_{\mathsf{PAEKS}})\\
&\quad \textit{If } b = b'\ \textit{then output } 1,\ \textit{and } 0\ \textit{otherwise.}
\end{aligned}
$$

Here, $\mathcal{O} := \{\mathcal{O}_C(\cdot,\cdot), \mathcal{O}_T(\cdot,\cdot)\}$. $\mathcal{O}_C$ takes $kw \in \mathcal{KS}$ and $i \in \{0,1\}$ as input, and returns the result of $\mathsf{PAEKS.Enc}(\mathsf{pk}^*_{\mathsf{R}[i]}, \mathsf{pk}_{\mathsf{S}}, \mathsf{sk}_{\mathsf{S}}, kw)$*. Here, there is no restriction. $\mathcal{O}_T$ takes $kw' \in \mathcal{KS}$ and $i \in \{0,1\}$ as input, and returns the result of* $\mathsf{PAEKS.Trapdoor}(\mathsf{pk}^*_{\mathsf{R}[i]}, \mathsf{pk}_{\mathsf{S}}, \mathsf{sk}_{\mathsf{R}[i]}, kw')$*. Here $(kw', i) \notin \{(kw^*_0, 0), (kw^*_1, 0), (kw^*_0, 1), (kw^*_1, 1)\}$. We say that a PAEKS scheme* PAEKS *is IND-CKA secure if the advantage*

$$\mathsf{Adv}^{\mathsf{IND\text{-}CKA}}_{\mathsf{PAEKS},\mathcal{A}}(\lambda) := |\Pr[\mathsf{Exp}^{\mathsf{IND\text{-}CKA}}_{\mathsf{PAEKS},\mathcal{A}}(\lambda) = 1] - 1/2|$$

is negligible in the security parameter λ.

Next, we define indistinguishability against the inside keyword guessing attack (IND-IKGA) which ensures that no information about the keyword is leaked from trapdoors. We also capture trapdoor anonymity simultaneously, i.e., $\mathsf{pk}^*_{\mathsf{R}[b]}$ is used for generating the challenge trapdoor.

Definition 5 (IND-IKGA)**.** *For all PPT adversaries $\mathcal{A}$, we define the following experiment.*

$$
\begin{aligned}
&\mathsf{Exp}^{\mathsf{IND\text{-}IKGA}}_{\mathsf{PAEKS},\mathcal{A}}(\lambda):\\
&\quad \mathsf{pp} \leftarrow \mathsf{PAEKS.Setup}(1^\lambda)\\
&\quad (\mathsf{pk}^*_{\mathsf{R}[0]}, \mathsf{sk}^*_{\mathsf{R}[0]}) \leftarrow \mathsf{PAEKS.KG_R}(\mathsf{pp});\ (\mathsf{pk}^*_{\mathsf{R}[1]}, \mathsf{sk}^*_{\mathsf{R}[1]}) \leftarrow \mathsf{PAEKS.KG_R}(\mathsf{pp})\\
&\quad (\mathsf{pk_S}, \mathsf{sk_S}) \leftarrow \mathsf{PAEKS.KG_S}(\mathsf{pp})\\
&\quad (kw^*_0, kw^*_1, \mathsf{state}) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{pp}, \mathsf{pk}^*_{\mathsf{R}[0]}, \mathsf{pk}^*_{\mathsf{R}[1]}, \mathsf{pk_S})\ \mathit{s.t.}\ kw^*_0, kw^*_1 \in \mathcal{KS} \wedge\ kw^*_0 \neq kw^*_1\\
&\quad b \xleftarrow{\$} \{0,1\};\ \mathsf{td}^*_{\mathsf{S},kw^*_b} \leftarrow \mathsf{PAEKS.Trapdoor}(\mathsf{pk}^*_{\mathsf{R}[b]}, \mathsf{pk_S}, \mathsf{sk}^*_{\mathsf{R}[b]}, kw^*_b)\\
&\quad b' \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{state}, \mathsf{td}^*_{\mathsf{S},kw^*_b})\\
&\quad \mathit{If}\ b = b'\ \mathit{then\ output}\ 1,\ \mathit{and}\ 0\ \mathit{otherwise.}
\end{aligned}
$$

Here, $\mathcal{O} := \{\mathcal{O}_C(\cdot,\cdot), \mathcal{O}_T(\mathsf{pk_R}, \cdot, \mathsf{sk_R}, \cdot)\}$. $\mathcal{O}_C$ takes $kw \in \mathcal{KS}$ and $i \in \{0,1\}$ as input, and returns the result of $\mathsf{PAEKS.Enc}(\mathsf{pk}^*_{\mathsf{R}[i]}, \mathsf{pk_S}, \mathsf{sk_S}, kw)$. Here, $(kw, i) \notin \{(kw^*_0, 0), (kw^*_1, 0), (kw^*_0, 1), (kw^*_1, 1)\}$. $\mathcal{O}_T$ takes $kw' \in \mathcal{KS}$ and $i \in \{0,1\}$ as input, and returns the result of $\mathsf{PAEKS.Trapdoor}(\mathsf{pk}^*_{\mathsf{R}[i]}, \mathsf{pk_S}, \mathsf{sk}_{\mathsf{R}[i]}, kw')$. Here $(kw', i) \notin \{(kw^*_0, 0), (kw^*_1, 0), (kw^*_0, 1), (kw^*_1, 1)\}$. We say that a PAEKS scheme PAEKS is IND-IKGA secure if the advantage

$$\mathsf{Adv}^{\mathsf{IND\text{-}IKGA}}_{\mathsf{PAEKS},\mathcal{A}}(\lambda) := |\Pr[\mathsf{Exp}^{\mathsf{IND\text{-}IKGA}}_{\mathsf{PAEKS},\mathcal{A}}(\lambda) = 1] - 1/2|$$

is negligible in the security parameter λ.

3 Definitions of BAEKS

In this section, we introduce the definitions of BAEKS. We mainly follow the definitions given in [27] but modify them to capture adaptive corruptions.

Definition 6 (Syntax of BAEKS)**.** *A BAEKS scheme BAEKS consists of the following six algorithms ($\mathsf{BAEKS.Setup}, \mathsf{BAEKS.KG_R}, \mathsf{BAEKS.KG_S}, \mathsf{BAEKS.Enc}, \mathsf{BAEKS.Trapdoor}, \mathsf{BAEKS.Test}$) defined as follows.*

$\mathsf{BAEKS.Setup}$**:** *The setup algorithm takes a security parameter λ and the maximum number of receivers N_{max} as input, and outputs a common parameter pp. We assume that pp implicitly contains the keyword space $\mathcal{KS}$.*

$\mathsf{BAEKS.KG_R}$**:** *The receiver key generation algorithm takes pp as input, and outputs a public key $\mathsf{pk_R}$ and secret key $\mathsf{sk_R}$.*

$\mathsf{BAEKS.KG_S}$**:** *The sender key generation algorithm takes pp as input, and outputs a public key $\mathsf{pk_S}$ and secret key $\mathsf{sk_S}$.*

$\mathsf{BAEKS.Enc}$**:** *The keyword encryption algorithm takes pp, a set of receivers $S = \{\mathsf{pk}_{\mathsf{R}[i]}\}_{i\in[1,N]}$ where $N \leq N_{\mathsf{max}}$, $\mathsf{pk_S}$, $\mathsf{sk_S}$, and a keyword $kw \in \mathcal{KS}$ as input, and outputs a ciphertext $\mathsf{ct_{PAEKS}}$.*

$\mathsf{BAEKS.Trapdoor}$**:** *The trapdoor algorithm takes $\mathsf{pk_R}$, $\mathsf{pk_S}$, $\mathsf{sk_R}$, and a keyword $kw' \in \mathcal{KS}$ as input, and outputs a trapdoor $\mathsf{td}_{\mathsf{R},kw'}$.*

$\mathsf{BAEKS.Test}$: *The test algorithm takes* $\mathsf{ct}_{\mathsf{BAEKS}}$ *and* $\mathsf{td}_{\mathsf{R},kw'}$ *as input, and outputs 1 or 0.*

Next, we define computational correctness, which ensures that the test algorithm outputs 1 if (1) the same keyword is specified when a ciphertext and a trapdoor are generated and (2) the trapdoor is generated by a receiver secret key, and the receiver public key is contained in a set of receivers, which is specified when the ciphertext is generated. The reason behind employing a computational concept here is that the correctness of the proposed generic construction relies on computational consistency (in a multi-receiver setting) of the underlying PAEKS scheme.

Definition 7 (Computational Correctness)**.** *For all PPT adversaries $\mathcal{A}$, we define the following experiment.*

$$
\begin{aligned}
&\mathsf{Exp}^{\mathsf{correct}}_{\mathsf{BAEKS},\mathcal{A}}(\lambda):\\
&\quad \mathsf{pp} \leftarrow \mathsf{BAEKS.Setup}(1^\lambda, N_{\mathsf{max}})\\
&\quad \textit{For } i \in [1, N_{\mathsf{max}}],\ (\mathsf{pk}_{\mathsf{R}[i]}, \mathsf{sk}_{\mathsf{R}[i]}) \leftarrow \mathsf{BAEKS.KG_R}(\mathsf{pp})\\
&\quad (\mathsf{pk_S}, \mathsf{sk_S}) \leftarrow \mathsf{BAEKS.KG_S}(\mathsf{pp})\\
&\quad (kw, S, \mathsf{pk_R}) \leftarrow \mathcal{A}(\mathsf{pp}, \{\mathsf{pk}_{\mathsf{R}[i]}\}_{i\in[1,N_{\mathsf{max}}]}, \mathsf{pk_S})\ \textit{ s.t. } kw \in \mathcal{KS} \wedge S \subseteq \{\mathsf{pk}_{\mathsf{R}[1]}, \dots, \mathsf{pk}_{\mathsf{R}[N_{\mathsf{max}}]}\} \wedge \mathsf{pk_R} \in S\\
&\quad \mathsf{ct}_{\mathsf{BAEKS}} \leftarrow \mathsf{BAEKS.Enc}(\mathsf{pp}, S, \mathsf{pk_S}, \mathsf{sk_S}, kw);\ \mathsf{td}_{\mathsf{R},kw} \leftarrow \mathsf{BAEKS.Trapdoor}(\mathsf{pk_R}, \mathsf{pk_S}, \mathsf{sk_R}, kw)\\
&\quad \textit{If } \mathsf{BAEKS.Test}(\mathsf{ct}_{\mathsf{BAEKS}}, \mathsf{td}_{\mathsf{R},kw}) = 1,\ \textit{then output } 1,\ \textit{and } 0 \textit{ otherwise.}
\end{aligned}
$$

We say that a BAEKS scheme BAEKS *is correct if the advantage*

$$\mathsf{Adv}^{\mathsf{correct}}_{\mathsf{BAEKS},\mathcal{A}}(\lambda) := \Pr[\mathsf{Exp}^{\mathsf{correct}}_{\mathsf{BAEKS},\mathcal{A}}(\lambda) = 1]$$

is negligible in the security parameter λ.

Next, we define computational consistency which ensures that the test algorithm outputs 0 if (1) different keywords are specified when a ciphertext and a trapdoor are generated, respectively, or (2) the trapdoor is generated by a receiver's secret key but the receiver's public key is not contained in a set of receivers which is specified when the ciphertext is generated. Especially, if $\mathsf{pk_R} \notin S$, then $\mathsf{BAEKS.Test}(\mathsf{ct}_{\mathsf{BAEKS}}, \mathsf{td}_{\mathsf{R},kw'}) = 0$ holds even if $kw = kw'$, where $\mathsf{ct}_{\mathsf{BAEKS}} \leftarrow \mathsf{BAEKS.Enc}(\mathsf{pp}, S, \mathsf{pk_S}, \mathsf{sk_S}, kw)$ and $\mathsf{td}_{\mathsf{R},kw'} \leftarrow \mathsf{BAEKS.Trapdoor}(\mathsf{pk_R}, \mathsf{pk_S}, \mathsf{sk_R}, kw')$. The following definition captures this case by the condition $(kw \neq kw' \vee \mathsf{pk_R} \notin S)$.

Definition 8 (Computational Consistency)**.** *For all PPT adversaries $\mathcal{A}$, we define the following experiment.*

$$
\begin{aligned}
&\mathsf{Exp}^{\mathsf{consist}}_{\mathsf{BAEKS},\mathcal{A}}(\lambda):\\
&\quad \mathsf{pp} \leftarrow \mathsf{BAEKS.Setup}(1^\lambda, N_{\mathsf{max}})\\
&\quad \textit{For } i \in [1, N_{\mathsf{max}}],\ (\mathsf{pk}_{\mathsf{R}[i]}, \mathsf{sk}_{\mathsf{R}[i]}) \leftarrow \mathsf{BAEKS.KG_R}(\mathsf{pp})\\
&\quad (\mathsf{pk_S}, \mathsf{sk_S}) \leftarrow \mathsf{BAEKS.KG_S}(\mathsf{pp})\\
&\quad (kw, kw', S, \mathsf{pk_R}) \leftarrow \mathcal{A}(\mathsf{pp}, \{\mathsf{pk}_{\mathsf{R}[i]}\}_{i\in[1,N_{\mathsf{max}}]}, \mathsf{pk_S})\\
&\qquad \textit{s.t. } kw, kw' \in \mathcal{KS} \wedge S \subseteq \{\mathsf{pk}_{\mathsf{R}[1]}, \dots, \mathsf{pk}_{\mathsf{R}[N_{\mathsf{max}}]}\} \wedge (kw \neq kw' \vee \mathsf{pk_R} \notin S)\\
&\quad \mathsf{ct}_{\mathsf{BAEKS}} \leftarrow \mathsf{BAEKS.Enc}(\mathsf{pp}, S, \mathsf{pk_S}, \mathsf{sk_S}, kw)\\
&\quad \mathsf{td}_{\mathsf{R},kw'} \leftarrow \mathsf{BAEKS.Trapdoor}(\mathsf{pk_R}, \mathsf{pk_S}, \mathsf{sk_R}, kw')\\
&\quad \textit{If } \mathsf{BAEKS.Test}(\mathsf{ct}_{\mathsf{BAEKS}}, \mathsf{td}_{\mathsf{R},kw'}) = 1,\ \textit{then output } 1,\ \textit{and } 0 \textit{ otherwise.}
\end{aligned}
$$

We say that a BAEKS scheme BAEKS *is consistent if the advantage*

$$\mathsf{Adv}^{\mathsf{consist}}_{\mathsf{BAEKS},\mathcal{A}}(\lambda) := \Pr[\mathsf{Exp}^{\mathsf{consist}}_{\mathsf{BAEKS},\mathcal{A}}(\lambda) = 1]$$

is negligible in the security parameter λ.

Next, we define indistinguishability against the chosen keyword attack (IND-CKA) which ensures that no information about the keyword is leaked from ciphertexts. We also capture ciphertext anonymity simultaneously, i.e., S_b^* is used for generating the challenge ciphertext. In our definition, adversaries $\mathcal{A}$ are allowed to obtain secret keys $\mathsf{sk}_{\mathsf{R}[i]}$. If $\mathsf{pk}_{\mathsf{R}[i]} \in S_0^* \cap S_1^*$, then $kw_0^* = kw_1^*$ is required to hold. Adversaries $\mathcal{A}$ are also allowed to obtain trapdoors generated by $\mathsf{sk}_{\mathsf{R}[i]}$. Similarly, if $\mathsf{pk}_{\mathsf{R}[i]} \in S_0^* \cap S_1^*$, then $kw_0^* = kw_1^*$ is required to hold.

Definition 9 (IND-CKA)**.** *For all PPT adversaries* $\mathcal{A}$*, we define the following experiment.*

$$\begin{aligned}
&\mathsf{Exp}^{\mathsf{IND\text{-}CKA}}_{\mathsf{BAEKS},\mathcal{A}}(\lambda):\\
&\quad \mathsf{pp} \leftarrow \mathsf{BAEKS.Setup}(1^\lambda, N_{\mathsf{max}})\\
&\quad \textit{For } i \in [1, N_{\mathsf{max}}],\ (\mathsf{pk}_{\mathsf{R}[i]}, \mathsf{sk}_{\mathsf{R}[i]}) \leftarrow \mathsf{BAEKS.KG_R}(\mathsf{pp})\\
&\quad (\mathsf{pk_S}, \mathsf{sk_S}) \leftarrow \mathsf{BAEKS.KG_S}(\mathsf{pp})\\
&\quad (kw_0^*, kw_1^*, S_0^*, S_1^*, \mathsf{state}) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{pp}, \{\mathsf{pk}_{\mathsf{R}[i]}\}_{i\in[1,N_{\mathsf{max}}]}, \mathsf{pk_S})\\
&\qquad \textit{s.t. } kw_0^*, kw_1^* \in \mathcal{KS} \wedge S_0^*, S_1^* \subseteq \{\mathsf{pk}_{\mathsf{R}[1]}, \ldots, \mathsf{pk}_{\mathsf{R}[N_{\mathsf{max}}]}\} \wedge |S_0^*| = |S_1^*|\\
&\quad b \xleftarrow{\$} \{0,1\};\ \mathsf{ct}^*_{\mathsf{BAEKS}} \leftarrow \mathsf{BAEKS.Enc}(\mathsf{pp}, S_b^*, \mathsf{pk_S}, \mathsf{sk_S}, kw_b^*)\\
&\quad b' \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{state}, \mathsf{ct}^*_{\mathsf{BAEKS}})\\
&\quad \textit{If } b = b' \textit{ then output } 1,\ \textit{and } 0 \textit{ otherwise.}
\end{aligned}$$

Here, $\mathcal{O} := \{\mathcal{O}_C(\cdot,\cdot), \mathcal{O}_T(\cdot,\cdot), \mathcal{O}_{Ext}(\cdot)\}$. $\mathcal{O}_C$ *takes* $kw \in \mathcal{KS}$ *and* $S \subseteq \{\mathsf{pk}_{\mathsf{R}[1]}, \ldots, \mathsf{pk}_{\mathsf{R}[N_{\mathsf{max}}]}\}$ *as input, and returns the result of* $\mathsf{BAEKS.Enc}(\mathsf{pp}, S, \mathsf{pk_S}, \mathsf{sk_S}, kw)$. *Here, there is no restriction.* $\mathcal{O}_T$ *takes* $kw' \in \mathcal{KS}$ *and* $\mathsf{pk}_{\mathsf{R}[i]} \in \{\mathsf{pk}_{\mathsf{R}[1]}, \ldots, \mathsf{pk}_{\mathsf{R}[N_{\mathsf{max}}]}\}$ *as input, and returns the result of* $\mathsf{BAEKS.Trapdoor}(\mathsf{pk}_{\mathsf{R}[i]}, \mathsf{pk_S}, \mathsf{sk}_{\mathsf{R}[i]}, kw')$. *Here, either* $kw' \notin \{kw_0^*, kw_1^*\}$ *or* $\mathsf{pk}_{\mathsf{R}[i]} \in S$ *where* $S \cap (S_0^* \triangle S_1^*) = \emptyset$. *If* $\mathsf{pk}_{\mathsf{R}[i]} \in S_0^* \cap S_1^*$*, then* $kw_0^* = kw_1^*$. $\mathcal{O}_{Ext}$ *takes* $\mathsf{pk}_{\mathsf{R}[i]} \in \{\mathsf{pk}_{\mathsf{R}[1]}, \ldots, \mathsf{pk}_{\mathsf{R}[N_{\mathsf{max}}]}\}$ *as input, and returns* $\mathsf{sk}_{\mathsf{R}[i]}$. *Here,* $\mathsf{pk}_{\mathsf{R}[i]} \in S$ *where* $S \cap (S_0^* \triangle S_1^*) = \emptyset$. *If* $\mathsf{pk}_{\mathsf{R}[i]} \in S_0^* \cap S_1^* \wedge kw' \in \{kw_0^*, kw_1^*\}$*, then* $kw_0^* = kw_1^*$. *We say that a BAEKS scheme* BAEKS *is IND-CKA secure if the advantage*

$$\mathsf{Adv}^{\mathsf{IND\text{-}CKA}}_{\mathsf{BAEKS},\mathcal{A}}(\lambda) := |\Pr[\mathsf{Exp}^{\mathsf{IND\text{-}CKA}}_{\mathsf{BAEKS},\mathcal{A}}(\lambda) = 1] - 1/2|$$

is negligible in the security parameter λ.

Next, we define indistinguishability against the inside keyword guessing attack (IND-IKGA) which ensures that no information about the keyword is leaked from trapdoors. We also capture trapdoor anonymity simultaneously, i.e., $(\mathsf{pk}^*_{\mathsf{R}[b]}, \mathsf{sk}^*_{\mathsf{R}[b]})$ are used for generating the challenge trapdoor.

Definition 10 (IND-IKGA)**.** *For all PPT adversaries $\mathcal{A}$, we define the following experiment.*

$$
\begin{aligned}
&\mathsf{Exp}^{\mathsf{IND\text{-}IKGA}}_{\mathsf{BAEKS},\mathcal{A}}(\lambda):\\
&\quad \mathsf{pp} \leftarrow \mathsf{BAEKS.Setup}(1^{\lambda}, N_{\mathsf{max}})\\
&\quad \textit{For } i \in [1, N_{\mathsf{max}}],\ (\mathsf{pk}_{\mathsf{R}[i]}, \mathsf{sk}_{\mathsf{R}[i]}) \leftarrow \mathsf{BAEKS.KG_R}(\mathsf{pp})\\
&\quad (\mathsf{pk_S}, \mathsf{sk_S}) \leftarrow \mathsf{BAEKS.KG_S}(\mathsf{pp})\\
&\quad (kw_0^*, kw_1^*, \mathsf{pk}^*_{\mathsf{R}[0]}, \mathsf{pk}^*_{\mathsf{R}[1]}, \mathsf{state}) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{pp}, \{\mathsf{pk}_{\mathsf{R}[i]}\}_{i\in[1,N_{\mathsf{max}}]}, \mathsf{pk_S})\\
&\quad\ \ s.t.\ kw_0^*, kw_1^* \in \mathcal{KS} \wedge\ kw_0^* \neq kw_1^* \wedge \mathsf{pk}^*_{\mathsf{R}[0]}, \mathsf{pk}^*_{\mathsf{R}[1]} \in \{\mathsf{pk}_{\mathsf{R}[i]}\}_{i\in[1,N_{\mathsf{max}}]}\\
&\quad b \xleftarrow{\$} \{0,1\};\ \mathsf{td}^*_{\mathsf{R}[b],kw_b^*} \leftarrow \mathsf{BAEKS.Trapdoor}(\mathsf{pk}^*_{\mathsf{R}[b]}, \mathsf{pk_S}, \mathsf{sk}^*_{\mathsf{R}[b]}, kw_b^*)\\
&\quad b' \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{state}, \mathsf{td}^*_{\mathsf{R}[b],kw_b^*})\\
&\quad \textit{If } b = b' \textit{ then output } 1,\ \textit{and } 0\ \textit{otherwise.}
\end{aligned}
$$

Here, $\mathcal{O} := \{\mathcal{O}_C(\cdot,\cdot), \mathcal{O}_T(\cdot,\cdot), \mathcal{O}_{Ext}(\cdot)\}$. $\mathcal{O}_C$ takes $kw \in \mathcal{KS}$ and $S \subseteq \{\mathsf{pk}_{\mathsf{R}[1]}, \dots, \mathsf{pk}_{\mathsf{R}[N_{\mathsf{max}}]}\}$ as input, and returns the result of $\mathsf{BAEKS.Enc}(\mathsf{pp}, S, \mathsf{pk_S}, \mathsf{sk_S}, kw)$. Here, either $kw' \notin \{kw_0^*, kw_1^*\}$ or $\mathsf{pk}^*_{\mathsf{R}[0]}, \mathsf{pk}^*_{\mathsf{R}[1]} \notin S$. $\mathcal{O}_T$ takes $kw' \in \mathcal{KS}$ and $\mathsf{pk}_{\mathsf{R}[i]} \in \{\mathsf{pk}_{\mathsf{R}[1]}, \dots, \mathsf{pk}_{\mathsf{R}[N_{\mathsf{max}}]}\}$ as input, and returns the result of $\mathsf{BAEKS.Trapdoor}(\mathsf{pk}_{\mathsf{R}[i]}, \mathsf{pk_S}, \mathsf{sk}_{\mathsf{R}[i]}, kw')$. Here, either $kw' \notin \{kw_0^*, kw_1^*\}$ or $\mathsf{pk}_{\mathsf{R}[i]} \notin \{\mathsf{pk}^*_{\mathsf{R}[0]}, \mathsf{pk}^*_{\mathsf{R}[1]}\}$. $\mathcal{O}_{Ext}$ takes $\mathsf{pk}_{\mathsf{R}[i]} \in \{\mathsf{pk}_{\mathsf{R}[1]}, \dots, \mathsf{pk}_{\mathsf{R}[N_{\mathsf{max}}]}\}$ as input, and returns $\mathsf{sk}_{\mathsf{R}[i]}$. Here, $\mathsf{pk}_{\mathsf{R}[i]} \notin \{\mathsf{pk}^*_{\mathsf{R}[0]}, \mathsf{pk}^*_{\mathsf{R}[1]}\}$. We say that a BAEKS scheme BAEKS is IND-IKGA secure if the advantage

$$\mathsf{Adv}^{\mathsf{IND\text{-}IKGA}}_{\mathsf{BAEKS},\mathcal{A}}(\lambda) := |\Pr[\mathsf{Exp}^{\mathsf{IND\text{-}IKGA}}_{\mathsf{BAEKS},\mathcal{A}}(\lambda) = 1] - 1/2|$$

is negligible in the security parameter λ.

4 Proposed Generic Construction

In this section, we demonstrate the proposed generic construction of BAEKS derived from PAEKS and a random permutation. Let $\mathsf{PAEKS} = (\mathsf{PAEKS.Setup}, \mathsf{PAEKS.KG_R}, \mathsf{PAEKS.KG_S}, \mathsf{PAEKS.Enc}, \mathsf{PAEKS.Trapdoor}, \mathsf{PAEKS.Test})$ be a PAEKS scheme. Let $\pi : \{1, \dots, N\} \to \{1, \dots, N\}$ be a random permutation for any $N \leq N_{\mathsf{max}}$. Intuitively, a BAEKS ciphertext is a set of PAEKS ciphertexts with each public key $\mathsf{pk}_{\mathsf{R}[i]}$ and the same keyword kw. Due to consistency in the multi-receiver setting, the test algorithm of the underlying PAEKS scheme outputs 0 for a ciphertext encrypted by $\mathsf{pk}_{\mathsf{R}[i]}$ and a trapdoor generated by $\mathsf{pk}_{\mathsf{R}[j]}$ and $i \neq j$, even if kw is associated to the trapdoor. That is, consistency in the multi-receiver setting acts as robustness in the generic construction of anonymous broadcast encryption. Moreover, a BAEKS ciphertext $(\mathsf{ct}_{\mathsf{PAEKS}1}, \dots, \mathsf{ct}_{\mathsf{PAEKS}N})$ is randomly sorted by a random permutation π such that $(\mathsf{ct}_{\mathsf{PAEKS}\pi(1)}, \dots, \mathsf{ct}_{\mathsf{PAEKS}\pi(N)})$. Thus, no information about receiver is revealed at least from the order of ciphertexts. The construction of a BAEKS scheme BAEKS from PAEKS is described below.

The Proposed Generic Construction

$\mathsf{BAEKS.Setup}(\lambda, N_{\mathsf{max}})$**:** Run $\mathsf{pp}' \leftarrow \mathsf{PAEKS.Setup}(1^{\lambda})$ and output $\mathsf{pp} = (\mathsf{pp}', N_{\mathsf{max}})$.

$\mathsf{BAEKS.KG_R}(\mathsf{pp})$**:** Parse $\mathsf{pp} = (\mathsf{pp}', N_{\mathsf{max}})$. Run $(\mathsf{pk_R}, \mathsf{sk_R}) \leftarrow \mathsf{PAEKS.KG_R}(\mathsf{pp}')$ and output $(\mathsf{pk_R}, \mathsf{sk_R})$.

$\mathsf{BAEKS.KG_S}(\mathsf{pp})$**:** Parse $\mathsf{pp} = (\mathsf{pp}', N_{\mathsf{max}})$. Run $(\mathsf{pk_S}, \mathsf{sk_S}) \leftarrow \mathsf{PAEKS.KG_S}(\mathsf{pp})$ and output $(\mathsf{pk_S}, \mathsf{sk_S})$.

$\mathsf{BAEKS.Enc}(\mathsf{pp}, S, \mathsf{pk_S}, \mathsf{sk_S}, kw)$**:** Parse $\mathsf{pp} = (\mathsf{pp}', N_{\mathsf{max}})$. Without loss of generality, we denote $S = \{\mathsf{pk}_{\mathsf{R}[1]}, \ldots, \mathsf{pk}_{\mathsf{R}[N]}\}$. For all $i \in [1, N_{\mathsf{max}}]$, run $\mathsf{ct}_{\mathsf{PAEKS}i} \leftarrow \mathsf{PAEKS.Enc}(\mathsf{pk}_{\mathsf{R}[i]}, \mathsf{pk_S}, \mathsf{sk_S}, kw)$. Output $\mathsf{ct_{BAEKS}} = \{\mathsf{ct}_{\mathsf{PAEKS}\pi(i)}\}_{i\in[1,N]}$.

$\mathsf{BAEKS.Trapdoor}(\mathsf{pk_R}, \mathsf{pk_S}, \mathsf{sk_R}, kw')$**:** Run $\mathsf{td}_{\mathsf{R},kw'} \leftarrow \mathsf{PAEKS.Trapdoor}(\mathsf{pk_R}, \mathsf{pk_S}, \mathsf{sk_R}, kw)$ and output $\mathsf{td}_{\mathsf{R},kw'}$.

$\mathsf{BAEKS.Test}(\mathsf{ct_{BAEKS}}, \mathsf{td}_{\mathsf{R},kw'})$**:** Parse $\mathsf{ct_{BAEKS}} = \{\mathsf{ct}_{\mathsf{PAEKS}i}\}_{i\in[1,N]}$. Output 1 if there exists $i \in [1, N]$ such that $\mathsf{PAEKS.Test}(\mathsf{ct}_{\mathsf{PAEKS}i}, \mathsf{td}_{\mathsf{R},kw'}) = 1$, and 0 otherwise.

Because of consistency in the multi-receiver setting of PAEKS, $\mathsf{PAEKS.Test}(\mathsf{ct}_{\mathsf{PAEKS}i}, \mathsf{td}_{\mathsf{R}[j],kw'}) = 0$ for $\mathsf{ct}_{\mathsf{PAEKS}i} \leftarrow \mathsf{PAEKS.Enc}(\mathsf{pk}_{\mathsf{R}[i]}, \mathsf{pk_S}, \mathsf{sk_S}, kw)$ and $\mathsf{td}_{\mathsf{R}[j],kw'} \leftarrow \mathsf{PAEKS.Trapdoor}(\mathsf{pk}_{\mathsf{R}[j]}, \mathsf{pk_S}, \mathsf{sk}_{\mathsf{R}[j]}, kw')$ if $(kw, i) \neq (kw', j)$. Thus, the proposed construction is correct. Note that the $\mathsf{BAEKS.Test}$ algorithm outputs 1 only if there exists one $i \in [1, N]$ such that $\mathsf{PAEKS.Test}(\mathsf{ct}_{\mathsf{PAEKS}i}, \mathsf{td}_{\mathsf{R},kw'}) = 1$ holds. This requires a stronger consistency and the underlying PAEKS scheme needs to provide consistency in the multi-receiver setting, and thus correctness holds in a computational manner. If we just require correctness in a usual manner, i.e., the $\mathsf{BAEKS.Test}$ algorithm outputs 1 even if there exist two or more $i \in [1, N]$ such that $\mathsf{PAEKS.Test}(\mathsf{ct}_{\mathsf{PAEKS}i}, \mathsf{td}_{\mathsf{R},kw'}) = 1$ holds, then the proposed construction is correct in a statistical manner.

In addition to provide correctness, due to consistency in the multi-receiver setting of PAEKS, the proposed construction is consistent because the condition $\mathsf{pk_R} \notin S$ in $\mathsf{Exp}^{\mathsf{consist}}_{\mathsf{BAEKS},\mathcal{A}}(\lambda)$ is also captured.

5 Security Analysis

In this section, we prove the following theorems. We note that Libert et al. [26] proved the IND-CCA security of the generic construction of anonymous broadcast encryption by assuming that the underlying encryption scheme is (weakly) robust. This robustness is required to handle decryption queries, where the decryption result using a different secret key is non-$\perp$. Since we do not consider CCA security, we do not employ consistency to prove IND-CKA/IND-IKGA security here.

Theorem 1. *The proposed construction is IND-CKA secure if the underlying PAEKS scheme is IND-CKA secure.*

Proof. The proof uses a sequence of games, where an adversary is given an encryption of kw_0^* for S_0^* as the challenge ciphertext in the first game, and the adversary is given an encryption of kw_1^* for S_1^* as the challenge ciphertext in the last game. Let $|S_0^*| = |S_1^*| = N^* \leq N_{\mathsf{max}}$ and $\ell = |S_0^* \cap S_1^*|$.

Game 0: This game corresponds to the real game when the challenger's bit is $b = 0$. Let E_0 be the event that $\mathcal{A}$ outputs $b' = 0$.

Game k $(1 \leq k \leq \ell)$**:** From S_0^* and S_1^*, let us define two ordered indices sets $\bar{S}_0^* = \{\theta_1, \ldots, \theta_\ell, \theta_{\ell+1}, \ldots, \theta_{N^*}\}$ and $\bar{S}_1^* = \{\rho_1, \ldots, \rho_\ell, \rho_{\ell+1}, \ldots, \rho_{N^*}\}$, where $\theta_i = \rho_i$ for $i \in [1, \ell]$ and $\theta_i \neq \rho_i$ for $i \in [\ell + 1, N^*]$. The challenge ciphertext $\mathsf{ct}^*_{\mathsf{BAEKS}}$ is generated as follows.

- For $j \in [1, k]$, compute $\mathsf{ct}_{\mathsf{PAEKS}j} \leftarrow \mathsf{PAEKS.Enc}(\mathsf{pk}_{\mathsf{R}[\theta_j]}, \mathsf{pk_S}, \mathsf{sk_S}, kw_1^*)$.
- For $j \in [k + 1, N^*]$, compute $\mathsf{ct}_{\mathsf{PAEKS}j} \leftarrow \mathsf{PAEKS.Enc}(\mathsf{pk}_{\mathsf{R}[\theta_j]}, \mathsf{pk_S}, \mathsf{sk_S}, kw_0^*)$.

Then, $\mathsf{ct}^*_{\mathsf{BAEKS}} = \{\mathsf{ct}_{\mathsf{PAEKS}\pi(i)}\}_{i\in[1,N^*]}$. Let E_k be the event that $\mathcal{A}$ outputs $b' = 0$ in Game k.

Game k' ($\ell+1 \leq k' \leq N^*$)**:** From S_0^* and S_1^*, again let define two ordered indices sets $\bar{S}_0^* = \{\theta_1, \ldots, \theta_\ell, \theta_{\ell+1}, \ldots, \theta_{N^*}\}$ and $\bar{S}_1^* = \{\rho_1, \ldots, \rho_\ell, \rho_{\ell+1}, \ldots, \rho_{N^*}\}$ where $\theta_i = \rho_i$ for $i \in [1, \ell]$ and $\theta_i \neq \rho_i$ for $i \in [\ell+1, N^*]$. The challenge ciphertext $\mathsf{ct}^*_{\mathsf{BAEKS}}$ is generated as follows.

- For $j \in [1, k']$, compute $\mathsf{ct}_{\mathsf{PAEKS}_j} \leftarrow \mathsf{PAEKS.Enc}(\mathsf{pk}_{\mathsf{R}[\rho_j]}, \mathsf{pk}_\mathsf{S}, \mathsf{sk}_\mathsf{S}, kw_1^*)$.
- For $j \in [k'+1, N^*]$, compute $\mathsf{ct}_{\mathsf{PAEKS}_j} \leftarrow \mathsf{PAEKS.Enc}(\mathsf{pk}_{\mathsf{R}[\theta_j]}, \mathsf{pk}_\mathsf{S}, \mathsf{sk}_\mathsf{S}, kw_0^*)$.

Then, $\mathsf{ct}^*_{\mathsf{BAEKS}} = \{\mathsf{ct}_{\mathsf{PAEKS}_{\pi(i)}}\}_{i \in [1, N^*]}$. Let $E_{k'}$ be the event that $\mathcal{A}$ outputs $b' = 0$ in Game k'.

Here, Game N^* corresponds to the real game when the challenger's bit is $b = 1$. We prove the following Lemma 1 and Lemma 2.

Lemma 1. *For each $k \in [1, \ell]$, Game k is indistinguishable from Game $k-1$ if the underlying PAEKS scheme is IND-CKA secure in the non-anonymous setting. Precisely, we can construct an algorithm $\mathcal{B}$ such that*

$$|\Pr[E_k] - \Pr[E_{k-1}]| \leq N_{\mathsf{max}} \cdot \mathsf{Adv}^{\mathsf{IND\text{-}CKA}}_{\mathsf{PAEKS},\mathcal{B}}(\lambda)$$

Proof. Let $\mathcal{A}$ be an adversary that distinguishes Game k and Game $k-1$. we construct an algorithm $\mathcal{B}$ that breaks the IND-CKA security of PAEKS as follows. Let $\mathcal{C}$ be the challenger of the IND-CKA security of PAEKS. For each $k \in [1, \ell]$, if $\mathcal{A}$ issues $\mathcal{O}_{Ext}(\mathsf{pk}_{\mathsf{R}[i]})$ such that $\mathsf{pk}_{\mathsf{R}[i]} \in S_0^* \cap S_1^*$, then $kw_0^* = kw_1^*$. Then, Game k and Game $k-1$ are identical. Thus, we can assume that $kw_0^* \neq kw_1^*$ and $\mathcal{A}$ does not issue $\mathcal{O}_{Ext}(\mathsf{pk}_{\mathsf{R}[i]})$ for $\mathsf{pk}_{\mathsf{R}[i]} \in S_0^* \cap S_1^*$.

$\mathcal{B}$ obtains $(\mathsf{pp}', \mathsf{pk}^*_\mathsf{R}, \mathsf{pk}_\mathsf{S})$ from $\mathcal{C}$. Recall that now non-anonymous setting is considered, $\mathsf{pk}^*_{\mathsf{R}[0]} = \mathsf{pk}^*_{\mathsf{R}[1]}$ and we set $\mathsf{pk}^*_\mathsf{R} = \mathsf{pk}^*_{\mathsf{R}[0]}$. $\mathcal{B}$ picks $i^* \xleftarrow{\$} \{1, N_{\mathsf{max}}\}$. For $i \in [1, N_{\mathsf{max}}] \setminus \{i^*\}$, $\mathcal{B}$ runs $(\mathsf{pk}_{\mathsf{R}[i]}, \mathsf{sk}_{\mathsf{R}[i]}) \leftarrow \mathsf{PAEKS.KG_R}(\mathsf{pp}')$. $\mathcal{B}$ sets $\mathsf{pp} = (\mathsf{pp}', N_{\mathsf{max}})$ and sends $(\mathsf{pp}, \{\mathsf{pk}_{\mathsf{R}[i]}\}_{i \in [1, N_{\mathsf{max}}]}, \mathsf{pk}_\mathsf{S})$ to $\mathcal{A}$.

- When $\mathcal{A}$ issues $\mathcal{O}_C(kw, S)$ where $|S| = N$, if $\mathsf{pk}^*_\mathsf{R} \in S$, then $\mathcal{B}$ issues $\mathcal{O}_C(kw, 0)$ of the underlying PAEKS scheme, obtains $\mathsf{ct}_{\mathsf{PAEKS}} \leftarrow \mathsf{PAEKS.Enc}(\mathsf{pk}^*_\mathsf{R}, \mathsf{pk}_\mathsf{S}, \mathsf{sk}_\mathsf{S}, kw)$, and sets $\mathsf{ct}_{\mathsf{PAEKS}_{i^*}} = \mathsf{ct}_{\mathsf{PAEKS}}$. $\mathcal{B}$ generates other PAEKS ciphertexts using $\mathsf{sk}_{\mathsf{R}[i]}$. $\mathcal{B}$ returns $\mathsf{ct}_{\mathsf{BAEKS}} = \{\mathsf{ct}_{\mathsf{PAEKS}_{\pi(i)}}\}_{i \in [1, N]}$ to $\mathcal{A}$.

- When $\mathcal{A}$ issues $\mathcal{O}_T(kw', \mathsf{pk}_{\mathsf{R}[i]})$, if $i = i^*$, then $\mathcal{B}$ issues $\mathcal{O}_T(kw', 0)$ of the underlying PAEKS scheme, obtains $\mathsf{td}_{\mathsf{R},kw'} \leftarrow \mathsf{PAEKS.Trapdoor}(\mathsf{pk}^*_\mathsf{R}, \mathsf{pk}_\mathsf{S}, \mathsf{sk}^*_\mathsf{R}, kw')$, and sends $\mathsf{td}_{\mathsf{R},kw'}$ to $\mathcal{A}$. If $i \neq i^*$, then $\mathcal{B}$ responds the query using $\mathsf{sk}_{\mathsf{R}[i]}$.

- When $\mathcal{A}$ issues $\mathcal{O}_{Ext}(\mathsf{pk}_{\mathsf{R}[i]})$ for $i \in [1, N_{\mathsf{max}}] \setminus \{i^*\}$, $\mathcal{B}$ returns $\mathsf{sk}_{\mathsf{R}[i]}$. When $\mathcal{A}$ issues $\mathcal{O}_{Ext}(\mathsf{pk}_{\mathsf{R}[i^*]})$, $\mathcal{B}$ aborts.

In the challenge phase, $\mathcal{A}$ declares $(kw_0^*, kw_1^*, S_0^*, S_1^*)$. $\mathcal{B}$ re-orders indices of S_0^* and S_1^* such that $\bar{S}_0^* = \{\theta_1, \ldots, \theta_\ell, \theta_{\ell+1}, \ldots, \theta_{N^*}\}$ and $\bar{S}_1^* = \{\rho_1, \ldots, \rho_\ell, \rho_{\ell+1}, \ldots, \rho_{N^*}\}$ where $\theta_i = \rho_i$ for $i \in [1, \ell]$ and $\theta_i \neq \rho_i$ for $i \in [\ell+1, N^*]$. If $\theta_k \neq i^*$, then $\mathcal{B}$ aborts. Here, we assume that $\theta_k = i^*$ holds with a probability of at least $1/N_{\mathsf{max}}$ since the choice of i^* is completely independent of $\mathcal{A}$'s view. We remark that if $\theta_k = i^*$, then $\mathsf{pk}_{\mathsf{R}[i^*]} = \mathsf{pk}^*_\mathsf{R} \in S_0^* \cap S_1^*$. Thus, $\mathcal{A}$ does not issue $\mathcal{O}_{Ext}(\mathsf{pk}_{\mathsf{R}[i^*]})$ as mentioned above. $\mathcal{B}$ sends (kw_0^*, kw_1^*) to $\mathcal{C}$ as the challenge keywords. $\mathcal{C}$ sends $\mathsf{ct}^*_{\mathsf{PAEKS}} \leftarrow \mathsf{PAEKS.Enc}(\mathsf{pk}^*_\mathsf{R}, \mathsf{pk}_\mathsf{S}, \mathsf{sk}_\mathsf{S}, kw_b^*)$ to $\mathcal{B}$ for some internally flipped random bit $b \xleftarrow{\$} \{0, 1\}$. The BAEKS challenge ciphertext $\mathsf{ct}^*_{\mathsf{BAEKS}} = \{\mathsf{ct}_{\mathsf{PAEKS}_{\pi(i)}}\}_{i \in [1, N^*]}$ is generated as follows.

- For $j \in [1, k-1]$, $\mathcal{B}$ issues $\mathcal{O}_C(kw_1^*, \mathsf{pk}_{\mathsf{R}[\theta_j]})$. Then $\mathcal{C}$ responds $\mathsf{ct}_{\mathsf{PAEKS}_j} \leftarrow \mathsf{PAEKS.Enc}(\mathsf{pk}_{\mathsf{R}[\theta_j]}, \mathsf{pk}_\mathsf{S}, \mathsf{sk}_\mathsf{S}, kw_1^*)$ to $\mathcal{B}$.

- For $j = k$, $\mathcal{B}$ sets $\mathsf{ct}_{\mathsf{PAEKS}j} = \mathsf{ct}^*_{\mathsf{PAEKS}}$.

- For $j \in [k{+}1, N^*]$, $\mathcal{B}$ issues $\mathcal{O}_C(kw^*_0, \mathsf{pk}_{\mathsf{R}[\theta_j]})$. Then $\mathcal{C}$ responds $\mathsf{ct}_{\mathsf{PAEKS}j} \leftarrow \mathsf{PAEKS.Enc}(\mathsf{pk}_{\mathsf{R}[\theta_j]}, \mathsf{pk}_\mathsf{S}, \mathsf{sk}_\mathsf{S}, kw^*_0)$ to $\mathcal{B}$.

$\mathcal{B}$ simulates $\mathcal{A}$'s queries as in the first phase. Finally, $\mathcal{A}$ outputs $b' \in \{0,1\}$, and $\mathcal{B}$ outputs the same result. If $\mathcal{C}$ chooses $b = 0$, then $\mathcal{B}$ is clearly playing Game $k-1$ whereas, if $b = 1$, $\mathcal{B}$ is playing Game k. This concludes the proof of Lemma 1. □

Lemma 2. *For each $k' \in [\ell{+}1, N^*]$, Game k' is indistinguishable from Game $k'{-}1$ if the underlying PAEKS scheme is IND-CKA secure. Precisely, we can construct an algorithm $\mathcal{B}$ such that*

$$|\Pr[E_{k'}] - \Pr[E_{k'-1}]| \leq N^2_{\mathsf{max}} \cdot \mathsf{Adv}^{\mathsf{IND\text{-}CKA}}_{\mathsf{PAEKS},\mathcal{B}}(\lambda)$$

Proof. Let $\mathcal{A}$ be an adversary that distinguishes Game k' and Game $k'-1$. we construct an algorithm $\mathcal{B}$ that breaks the IND-CKA security of PAEKS as follows. Let $\mathcal{C}$ be the challenger of the IND-CKA security of PAEKS.

$\mathcal{B}$ obtains $(\mathsf{pp}', \mathsf{pk}^*_{\mathsf{R}[0]}, \mathsf{pk}^*_{\mathsf{R}[1]}, \mathsf{pk}_\mathsf{S})$ from $\mathcal{C}$. $\mathcal{B}$ picks two distinct indices $i^*_0, i^*_1 \xleftarrow{\$} \{1, N_{\mathsf{max}}\}$ and sets $\mathsf{pk}_{\mathsf{R}[i^*_0]} = \mathsf{pk}^*_{\mathsf{R}[0]}$ and $\mathsf{pk}^*_{\mathsf{R}[i^*_1]} = \mathsf{pk}_{\mathsf{R}[1]}$. For $i \in [1, N_{\mathsf{max}}] \setminus \{i^*_0, i^*_1\}$, $\mathcal{B}$ runs $(\mathsf{pk}_{\mathsf{R}[i]}, \mathsf{sk}_{\mathsf{R}[i]}) \leftarrow \mathsf{PAEKS.KG_R}(\mathsf{pp}')$. $\mathcal{B}$ sets $\mathsf{pp} = (\mathsf{pp}', N_{\mathsf{max}})$ and sends $(\mathsf{pp}, \{\mathsf{pk}_{\mathsf{R}[i]}\}_{i \in [1, N_{\mathsf{max}}]}, \mathsf{pk}_\mathsf{S})$ to $\mathcal{A}$.

- When $\mathcal{A}$ issues $\mathcal{O}_C(kw, S)$ where $|S| = N$, if $\mathsf{pk}_{\mathsf{R}[i^*_0]} \in S$, then $\mathcal{B}$ issues $\mathcal{O}_C(kw, 0)$ of the underlying PAEKS scheme, obtains $\mathsf{ct}_{\mathsf{PAEKS}} \leftarrow \mathsf{PAEKS.Enc}(\mathsf{pk}^*_{\mathsf{R}[0]}, \mathsf{pk}_\mathsf{S}, \mathsf{sk}_\mathsf{S}, kw)$, and sets $\mathsf{ct}_{\mathsf{PAEKS}i^*_0} = \mathsf{ct}_{\mathsf{PAEKS}}$. If $\mathsf{pk}_{\mathsf{R}[i^*_1]} \in S$, then $\mathcal{B}$ issues $\mathcal{O}_C(kw, 1)$ of the underlying PAEKS scheme, obtains $\mathsf{ct}_{\mathsf{PAEKS}} \leftarrow \mathsf{PAEKS.Enc}(\mathsf{pk}^*_{\mathsf{R}[1]}, \mathsf{pk}_\mathsf{S}, \mathsf{sk}_\mathsf{S}, kw)$, and sets $\mathsf{ct}_{\mathsf{PAEKS}i^*_1} = \mathsf{ct}_{\mathsf{PAEKS}}$. $\mathcal{B}$ generates other PAEKS ciphertexts using $\mathsf{sk}_{\mathsf{R}[i]}$. $\mathcal{B}$ returns $\mathsf{ct}_{\mathsf{BAEKS}} = \{\mathsf{ct}_{\mathsf{PAEKS}\pi(i)}\}_{i \in [1,N]}$ to $\mathcal{A}$.

- When $\mathcal{A}$ issues $\mathcal{O}_T(kw', \mathsf{pk}_{\mathsf{R}[i]})$, if $i = i^*_0$, then $\mathcal{B}$ issues $\mathcal{O}_T(kw', 0)$ of the underlying PAEKS scheme, obtains $\mathsf{td}_{\mathsf{R},kw'} \leftarrow \mathsf{PAEKS.Trapdoor}(\mathsf{pk}^*_{\mathsf{R}[0]}, \mathsf{pk}_\mathsf{S}, \mathsf{sk}^*_{\mathsf{R}[0]}, kw')$, and sends $\mathsf{td}_{\mathsf{R},kw'}$ to $\mathcal{A}$. If $i = i^*_1$, then $\mathcal{B}$ issues $\mathcal{O}_T(kw', 1)$ of the underlying PAEKS scheme, obtains $\mathsf{td}_{\mathsf{R},kw'} \leftarrow \mathsf{PAEKS.Trapdoor}(\mathsf{pk}^*_{\mathsf{R}[1]}, \mathsf{pk}_\mathsf{S}, \mathsf{sk}^*_{\mathsf{R}[1]}, kw')$, and sends $\mathsf{td}_{\mathsf{R},kw'}$ to $\mathcal{A}$. If $i \notin \{i^*_0, i^*_1\}$, then $\mathcal{B}$ responds the query using $\mathsf{sk}_{\mathsf{R}[i]}$.

- When $\mathcal{A}$ issues $\mathcal{O}_{Ext}(\mathsf{pk}_{\mathsf{R}[i]})$ for $i \in [1, N_{\mathsf{max}}] \setminus \{i^*\}$, $\mathcal{B}$ returns $\mathsf{sk}_{\mathsf{R}[i]}$. When $\mathcal{A}$ issues $\mathcal{O}_{Ext}(\mathsf{pk}_{\mathsf{R}[i]})$ for $i \in \{i^*_0, i^*_1\}$, $\mathcal{B}$ aborts.

In the challenge phase, $\mathcal{A}$ declares $(kw^*_0, kw^*_1, S^*_0, S^*_1)$. $\mathcal{B}$ re-orders indices of S^*_0 and S^*_1 such that $\bar{S}^*_0 = \{\theta_1, \ldots, \theta_\ell, \theta_{\ell+1}, \ldots, \theta_{N^*}\}$ and $\bar{S}^*_1 = \{\rho_1, \ldots, \rho_\ell, \rho_{\ell+1}, \ldots, \rho_{N^*}\}$ where $\theta_i = \rho_i$ for $i \in [1, \ell]$ and $\theta_i \neq \rho_i$ for $i \in [\ell+1, N^*]$. If $\theta_{k'} \neq i^*_0$ or $\rho_{k'} \neq i^*_1$, then $\mathcal{B}$ aborts. Here, we assume $\theta_{k'} = i^*_0$ and $\rho_{k'} = i^*_1$, which holds with a probability of at least $1/N_{\mathsf{max}}(N_{\mathsf{max}} - 1) > 1/N^2_{\mathsf{max}}$ since the choice of (i^*_0, i^*_1) is completely independent of $\mathcal{A}$'s view. We remark that if $\theta_{k'} = i^*_0$ and $\rho_{k'} = i^*_1$, then $\mathsf{pk}_{\mathsf{R}[i^*_0]}, \mathsf{pk}_{\mathsf{R}[i^*_1]} \in S^*_0 \triangle S^*_1$ and thus $\mathcal{A}$ does not issue both $\mathcal{O}_{Ext}(\mathsf{pk}_{\mathsf{R}[i^*_0]})$ and $\mathcal{O}_{Ext}(\mathsf{pk}_{\mathsf{R}[i^*_1]})$. $\mathcal{B}$ sends (kw^*_0, kw^*_1) to $\mathcal{C}$ as the challenge keywords. $\mathcal{C}$ sends $\mathsf{ct}^*_{\mathsf{PAEKS}} \leftarrow \mathsf{PAEKS.Enc}(\mathsf{pk}^*_{\mathsf{R}[b]}, \mathsf{pk}_\mathsf{S}, \mathsf{sk}_\mathsf{S}, kw^*_b)$ to $\mathcal{B}$ for some internally flipped random bit $b \xleftarrow{\$} \{0,1\}$. The BAEKS challenge ciphertext $\mathsf{ct}^*_{\mathsf{BAEKS}} = \{\mathsf{ct}_{\mathsf{PAEKS}\pi(i)}\}_{i \in [1,N^*]}$ is generated as follows.

- For $j \in [1, k'{-}1]$, $\mathcal{B}$ issues $\mathcal{O}_C(kw^*_1, \mathsf{pk}_{\mathsf{R}[\rho_j]})$. Then $\mathcal{C}$ responds $\mathsf{ct}_{\mathsf{PAEKS}j} \leftarrow \mathsf{PAEKS.Enc}(\mathsf{pk}_{\mathsf{R}[\rho_j]}, \mathsf{pk}_\mathsf{S}, \mathsf{sk}_\mathsf{S}, kw^*_1)$ to $\mathcal{B}$.

- For $j = k'$, $\mathcal{B}$ sets $\mathsf{ct}_{\mathsf{PAEKS}j} = \mathsf{ct}^*_{\mathsf{PAEKS}}$.

- For $j \in [k'+1, N^*]$, $\mathcal{B}$ issues $\mathcal{O}_C(kw_0^*, \mathsf{pk}_{\mathsf{R}[\theta_j]})$. Then $\mathcal{C}$ responds $\mathsf{ct}_{\mathsf{PAEKS}j} \leftarrow \mathsf{PAEKS.Enc}(\mathsf{pk}_{\mathsf{R}[\theta_j]}, \mathsf{pk}_{\mathsf{S}}, \mathsf{sk}_{\mathsf{S}}, kw_0^*)$ to $\mathcal{B}$.

$\mathcal{B}$ simulates $\mathcal{A}$'s queries as in the first phase. Finally, $\mathcal{A}$ outputs $b' \in \{0,1\}$. and $\mathcal{B}$ outputs the same result. If $\mathcal{C}$ chooses $b=0$, then $\mathcal{B}$ is clearly playing Game $k'-1$ whereas, if $b=1$, $\mathcal{B}$ is playing Game k'. This concludes the proof of Lemma 2. $\square$

From Lemma 1 and Lemma 2, we have $|\Pr[E_0] - \Pr[E_{N^*}]| \leq \ell \cdot N_{\mathsf{max}} \cdot \mathsf{Adv}^{\mathsf{IND\text{-}CKA}}_{\mathsf{PAEKS},\mathcal{B}}(\lambda) + (N^* - \ell) \cdot N_{\mathsf{max}}^2 \cdot \mathsf{Adv}^{\mathsf{IND\text{-}CKA}}_{\mathsf{PAEKS},\mathcal{B}}(\lambda) \leq N_{\mathsf{max}}^3 \cdot \mathsf{Adv}^{\mathsf{IND\text{-}CKA}}_{\mathsf{PAEKS},\mathcal{B}}(\lambda)$. This concludes the proof of Theorem 1. $\square$

Theorem 2. *The proposed construction is IND-IKGA secure if the underlying PAEKS scheme is IND-IKGA secure.*

Proof Sketch. Since a BAEKS trapdoor is a PAEKS trapdoor in the proposed construction, the proof of Theorem 2 is straightforward. Let $\mathcal{A}$ be the adversary that breaks the IND-IKGA security. We construct an algorithm $\mathcal{B}$ that breaks the IND-IKGA security of the underlying PAEKS scheme. We need to consider that $\mathcal{B}$ embeds two public keys, say $(\mathsf{pk}^*_{\mathsf{R}[0]}, \mathsf{pk}^*_{\mathsf{R}[1]})$, given by the challenger of the IND-IKGA security of PAEKS $\mathcal{C}$, to $\{\mathsf{pk}_{\mathsf{R}[i]}\}_{i\in[1,N_{\mathsf{max}}]}$, and expects that $(\mathsf{pk}^*_{\mathsf{R}[0]}, \mathsf{pk}^*_{\mathsf{R}[1]})$ will be selected by $\mathcal{A}$ in the challenge phase. The guessing is correct with a probability of at least N_{max}^2. If the guess is correct, then $\mathcal{B}$ can simulate all queries issued by $\mathcal{A}$ by forwarding them to $\mathcal{C}$, and can break the IND-IKGA security of the underlying PAEKS scheme using $\mathcal{A}$. $\square$

6 Qin et al. PAEKS

In this section, we briefly explain that the Qin et al. PAEKS scheme [35] provides consistency in the multi-receiver setting and ciphertext anonymity, but it does not provide trapdoor anonymity. We emphasize that trapdoor anonymity is not required in the original PAEKS security definition. The Qin et al. PAEKS scheme is described as follows.

$\mathsf{PAEKS.Setup}(\lambda)$**:** Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear pairing where $\mathbb{G}$ and $\mathbb{G}_T$ be groups with prime order p and $\mathbb{G} = \langle g \rangle$. $H_1 : \{0,1\}^* \to \mathbb{G}$, $H_2 : \mathbb{G} \to \{0,1\}^\lambda$, and $H_3 : \mathbb{G} \to \{0,1\}^\lambda$ be hash functions which are modeled as random oracles. Output $\mathsf{pp} = (g, \mathbb{G}, \mathbb{G}_T, e, p, H_1, H_2, H_3)$.

$\mathsf{PAEKS.KG_R}(\mathsf{pp})$**:** Choose $x, v \xleftarrow{\$} \mathbb{Z}_p$. Output $\mathsf{pk_R} = (\mathsf{pk_R^{(1)}}, \mathsf{pk_R^{(2)}}) = (g^x, g^v)$ and $\mathsf{sk_R} = (\mathsf{sk_R^{(1)}}, \mathsf{sk_R^{(2)}}) = (x, v)$.

$\mathsf{PAEKS.KG_S}(\mathsf{pp})$**:** Choose $u \xleftarrow{\$} \mathbb{Z}_p$. output $\mathsf{pk_S} = g^u$ and $\mathsf{sk_S} = u$.

$\mathsf{PAEKS.Enc}(\mathsf{pk_R}, \mathsf{pk_S}, \mathsf{sk_S}, kw)$**:** Parse $\mathsf{pk_R} = (\mathsf{pk_R^{(1)}}, \mathsf{pk_R^{(2)}})$ and $\mathsf{sk_S} = u$. Choose $r \xleftarrow{\$} \mathbb{Z}_p$ and compute $A = g^x$. Compute $\mathsf{DHkey_{S,R}} = (\mathsf{pk_R^{(2)}})^u$ $(= g^{uv})$, $h = H_1(kw||\mathsf{pk_S}||\mathsf{pk_R}||H_3(\mathsf{DHkey_{S,R}}))$, and $B = H_2(e(h^r, \mathsf{pk_R^{(1)}})))$. Output $\mathsf{ct_{PAEKS}} = (A, B)$.

$\mathsf{PAEKS.Trapdoor}(\mathsf{pk_R}, \mathsf{pk_S}, \mathsf{sk_R}, kw')$**:** Parse $\mathsf{pk_R} = (\mathsf{pk_R^{(1)}}, \mathsf{pk_R^{(2)}})$ and $\mathsf{sk_R} = (\mathsf{sk_R^{(1)}}, \mathsf{sk_R^{(2)}})$. Compute $\mathsf{DHkey_{S,R}} = \mathsf{pk_S}^{\mathsf{sk_R^{(2)}}}$ $(= g^{uv})$ and $h' = H_1(kw'||\mathsf{pk_S}||\mathsf{pk_R}||H_3(\mathsf{DHkey_{S,R}}))$. Output $\mathsf{td}_{\mathsf{R},kw'} = (h')^{\mathsf{sk_R^{(1)}}} = (h')^x$.

$\mathsf{PAEKS.Test}(\mathsf{ct_{PAEKS}}, \mathsf{td}_{\mathsf{R},kw'})$**:** Parse $\mathsf{ct_{PAEKS}} = (A, B)$. Output 1 if $H_2(e(A, \mathsf{td}_{\mathsf{R},kw'})) = B$ and 0, otherwise.

Intuitively, a DH key $\mathsf{DHkey}_{\mathsf{S,R}} = (\mathsf{pk}_{\mathsf{R}}^{(2)})^{\mathsf{sk}_{\mathsf{S}}} = (\mathsf{pk}_{\mathsf{S}})^{\mathsf{sk}_{\mathsf{R}}^{(2)}} = g^{uv}$ is defined, which is fixed when a sender and a receiver are fixed. The value h is computed by a keyword to be encrypted and a DH key such that $h = H_1(kw||\mathsf{pk}_{\mathsf{S}}||\mathsf{pk}_{\mathsf{R}}||H_3(\mathsf{DHkey}_{\mathsf{S,R}}))$. Since H_1 is modeled as a random oracle, informally, no information about kw is revealed from h. Here, to formally prove the IND-IKGA security, H_3 is required. A ciphertext is $A = g^r$ and $B = H_2(e(h^r, \mathsf{pk}_{\mathsf{R}}^{(1)}))$ for $r \xleftarrow{\$} \mathbb{Z}_p$. Thus, informally, no information of kw is revealed from (A, B) since H_2 is modeled as a random oracle. Formally, Qin et al. proved the IND-CKA security under the bilinear Diffie-Hellman (BDH) assumption. Simultaneously, we observe that receiver information, i.e., pk_{R} is also not revealed from (A, B). Precisely, for two challenge keywords kw_0^* and kw_1^* and two receivers' public keys $\mathsf{pk}_{\mathsf{R}[0]}$ and $\mathsf{pk}_{\mathsf{R}[1]}$, the challenge bit b is hidden from $H_1(kw_b^*||\mathsf{pk}_{\mathsf{S}}||\mathsf{pk}_{\mathsf{R}[b]}||H_3(\mathsf{DHkey}_{\mathsf{S,R}b}))$ and the simulation given in [35] still works. The value h' is computed by a keyword to be searched and a DH key, such that $h' = H_1(kw'||\mathsf{pk}_{\mathsf{S}}||\mathsf{pk}_{\mathsf{R}}||H_3(\mathsf{DHkey}_{\mathsf{S,R}}))$, and $\mathsf{td}_{\mathsf{R},kw'} = (h')^x$. If $kw = kw'$ and the sender and the receiver are the same, then $h = h'$ holds. If $kw \neq kw'$ or either the sender or the receiver is different, then $h \neq h'$ holds due to the collision resistance of H_1. Thus, consistency in the multi-receiver setting holds. Since H_1 is modeled as random oracle, informally, no information of kw' is revealed from h' and thus no information of kw' is revealed from $\mathsf{td}_{\mathsf{R},kw'} = (h')^x$. Formally, Qin et al. introduced the computational oracle Diffie-Hellman (CODH) problem, and proved that the scheme provides the IND-IKGA security under the CODH assumption.

However, because $(g, h', \mathsf{pk}_{\mathsf{R}}^{(1)}, \mathsf{td}_{\mathsf{R},kw'}) = (g, h', g^x, (h')^x)$ is a decisional Diffie-Hellman (DDH) tuple, $e(\mathsf{pk}_{\mathsf{R}}^{(1)}, h') = e(g, \mathsf{td}_{\mathsf{R},kw'})$ holds if $\mathsf{td}_{\mathsf{R},kw'}$ is generated by the receiver (whose public key is pk_{R}). Thus, the Qin et al. PAEKS scheme does not provide trapdoor anonymity. To provide trapdoor anonymity, one may employ type-3 asymmetric pairings; where $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, and there is no efficiently computable isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$. Then, the DDH assumption holds over both $\mathbb{G}_1$ and $\mathbb{G}_2$. To prevent the DDH test, $(g, h', \mathsf{pk}_{\mathsf{R}}^{(1)}, \mathsf{td}_{\mathsf{R},kw'})$ must belong to the same group. However, a ciphertext consists of $B = H_2(e(h^r, \mathsf{pk}_{\mathsf{R}}^{(1)}))$, i.e., h and $\mathsf{pk}_{\mathsf{R}}^{(1)}$ belong to different groups, and thus h and h' also belong to different groups. This violates the correctness of the Qin et al. scheme that requires $h = h'$ if $kw = kw'$ and the sender and the receiver are the same. Thus, it seems nontrivial to provide trapdoor anonymity even if asymmetric pairings are employed.

7 Conclusion

In this paper, we propose a generic construction of BAEKS from PAEKS providing ciphertext and trapdoor anonymity and consistency in the multi-receiver setting. Our generic constructions provide adaptive corruptions.

The proposed construction requires approximately $|S|/2$-times PAEKS test procedures. To reduce the number of decryption attempts in the generic construction of anonymous broadcast encryption, Libert et al. [26] proposed an anonymous hint system that provides $O(1)$ decryption cost in terms of the number of cryptographic operations. Unfortunately, we could not directly employ this anonymous hint system because the test algorithm was run by a cloud server in BAEKS, whereas the decryption algorithm was run by a receiver in anonymous broadcast encryption. Thus, the cloud server could observe the secret key of the hint system. Because of ciphertext anonymity (which is implied by IND-CKA in our definition), it is required that the cloud server has no information about the receivers before running the test algorithm. That is, if a hint system can be employed, then the cloud server obtains information about the receivers before running the test algorithm. Consequently, we did not employ a hint system in this paper. We leave this task as an interesting future work.

Fazio et al. [18] also proposed a generic construction of anonymous broadcast encryption that provides outsider anonymity, where no information about a receiver is leaked from ciphertexts against outsiders, i.e., an adversary is allowed to obtain secret keys of outsiders who belong to a set S where $S \cap (S_0^* \cup S_1^*) = \emptyset$. Regarding the number of receivers, the Libert et al. construction provides a linear-size ciphertext, whereas the Fazio et al. construction provides a sublinear-size ciphertext using the subset cover framework [31] at the expense of a weak anonymity level. Although outsider anonymity seems sufficient in some applications, the construction proposed by Fazio et al. cannot be extended to BAEKS directly because Fazio et al. employed anonymous and weakly robust identity-based encryption. Recently, a generic construction of BEKS from anonymous and weakly robust 3-level hierarchical identity-based encryption has been proposed [17] but it does not consider authenticity. Employing the Fazio et al. construction in the BAEKS context is left as a future work.

Though Yao et al. [37] proposed a lattice-based PAEKS scheme, they did not define consistency and thus it is unclear whether the Yao et al. PAEKS scheme provides consistency in the multi-receiver setting. Moreover, they did not define trapdoor privacy (they considered ciphertext privacy that guarantees no information about keyword is revealed from ciphertexts, and considered unforgeability of ciphertexts and trapdoors). Thus, we do not consider the Yao et al. scheme as a building block of the proposed generic construction. Cheng and Meng [13] proposed a PAEKS scheme from LWE (learning with errors). In their security proof, almost all ciphertext components are switched to random values. However, one component is selected from the receiver public key-related distribution. Although it is sufficient to prove that no information about the keyword is revealed from ciphertexts, it is unclear whether the Cheng-Meng PAEKS scheme provides ciphertext anonymity. We leave this to be investigated in a future study.

Acknowledgment: This work was supported by JSPS KAKENHI Grant Number JP21K11897. The main part of this study was done when the author was with the National Institute of Information and Communications Technology (NICT).

References

[1] Michel Abdalla, Mihir Bellare, and Gregory Neven. Robust encryption. In *TCC*, pages 480–497, 2010.

[2] Michel Abdalla, Mihir Bellare, and Gregory Neven. Robust encryption. *Journal of Cryptology*, 31(2):307–350, 2018.

[3] Mohamed Ali, Hamza Ali, Ting Zhong, Fagen Li, Zhiguan Qin, and A. A. Ahmed Abdelrahaman. Broadcast searchable keyword encryption. In *IEEE CSE*, pages 1010–1016, 2014.

[4] Miguel Ambrona, Gilles Barthe, and Benedikt Schmidt. Generic transformations of predicate encodings: Constructions and applications. In *CRYPTO*, pages 36–66, 2017.

[5] Nuttapong Attrapadung, Jun Furukawa, and Hideki Imai. Forward-secure and searchable broadcast encryption with short ciphertexts and private keys. In *ASIACRYPT*, pages 161–177, 2006.

[6] Adam Barth, Dan Boneh, and Brent Waters. Privacy in encrypted content distribution using private broadcast encryption. In *Financial Cryptography and Data Security*, pages 52–64, 2006.

[7] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In *EUROCRYPT*, pages 506–522, 2004.

[8] Marco Calderini, Riccardo Longo, Massimiliano Sala, and Irene Villa. Searchable encryption with randomized ciphertext and randomized keyword search. *IACR Cryptol. ePrint Arch.*, page 945, 2022.

[9] Sanjit Chatterjee and Sayantan Mukherjee. Keyword search meets membership testing: Adaptive security from SXDH. In *INDOCRYPT*, pages 21–43, 2018.

[10] Jie Chen, Romain Gay, and Hoeteck Wee. Improved dual system ABE in prime-order groups via predicate encodings. In *EUROCRYPT*, pages 595–624, 2015.

[11] Jie Chen and Junqing Gong. ABE with tag made easy - concise framework and new instantiations in prime-order groups. In *ASIACRYPT*, pages 35–65, 2017.

[12] Leixiao Cheng and Fei Meng. Security analysis of Pan et al.'s "public-key authenticated encryption with keyword search achieving both multi-ciphertext and multi-trapdoor indistinguishability". *Journal of Systems Architecture*, 119:102248, 2021.

[13] Leixiao Cheng and Fei Meng. Public key authenticated encryption with keyword search from LWE. In *ESORICS*, pages 303–324, 2022.

[14] Tianyu Chi, Baodong Qin, and Dong Zheng. An efficient searchable public-key authenticated encryption for cloud-assisted medical internet of things. *Wireless Communications and Mobile Computing*, 2020:8816172:1–8816172:11, 2020.

[15] Keita Emura. Generic construction of public-key authenticated encryption with keyword search revisited: Stronger security and efficient construction. In *ACM APKC*, pages 39–49, 2022.

[16] Keita Emura. Generic construction of fully anonymous broadcast authenticated encryption with keyword search with adaptive corruptions. *IET Information Security*, 2023:9922828:1–9922828:12, 2023.

[17] Keita Emura, Kaisei Kajita, and Go Ohtake. Outsider-anonymous broadcast encryption with keyword search: Generic construction, CCA security, and with sublinear ciphertexts. *IACR Cryptology ePrint Archive*, page 1119, 2023.

[18] Nelly Fazio and Irippuge Milinda Perera. Outsider-anonymous broadcast encryption with sublinear ciphertexts. In *Public Key Cryptography*, pages 225–242, 2012.

[19] Tao Feng and Jiewen Si. Certificateless searchable encryption scheme in multi-user environment. *Cryptography*, 6(4):61, 2022.

[20] Qiong Huang and Hongbo Li. An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks. *Information Sciences*, 403:1–14, 2017.

[21] Peng Jiang, Fuchun Guo, and Yi Mu. Efficient identity-based broadcast encryption with keyword search against insider attacks for database systems. *Theoretical Computer Science*, 767:51–72, 2019.

[22] Aggelos Kiayias, Ozgur Oksuz, Alexander Russell, Qiang Tang, and Bing Wang. Efficient encrypted keyword search for multi-user data sharing. In *ESORICS*, pages 173–195, 2016.

[23] Aggelos Kiayias and Katerina Samari. Lower bounds for private broadcast encryption. In *Information Hiding*, pages 176–190, 2012.

[24] Hirokazu Kobayashi, Yohei Watanabe, and Junji Shikata. Asymptotically tight lower bounds in anonymous broadcast encryption and authentication. In *IMACC*, pages 105–128, 2021.

[25] Jiangtao Li and Junqing Gong. Improved anonymous broadcast encryptions - tight security and shorter ciphertext. In *ACNS*, pages 497–515, 2018.

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

[27] Xueqiao Liu, Kai He, Guomin Yang, Willy Susilo, Joseph Tonien, and Qiong Huang. Broadcast authenticated encryption with keyword search. In *ACISP*, pages 193–213, 2021.

[28] Zi-Yuan Liu, Yi-Fan Tseng, Raylin Tso, Masahiro Mambo, and Yu-Chi Chen. Public-key authenticated encryption with keyword search: Cryptanalysis, enhanced security, and quantum-resistant instantiation. In *ACM ASIACCS*, pages 423–436, 2022.

[29] Mimi Ma, Shuqin Fan, and Dengguo Feng. Multi-user certificateless public key encryption with conjunctive keyword search for cloud-based telemedicine. *Journal of Information Security and Applications*, 55:102652, 2020.

[30] Sayantan Mukherjee. Statistically consistent broadcast authenticated encryption with keyword search: Adaptive security from standard assumptions. In *ACISP*, pages 523–552, 2023.

[31] Dalit Naor, Moni Naor, and Jeffery Lotspiech. Revocation and tracing schemes for stateless receivers. In *CRYPTO*, pages 41–62, 2001.

[32] Mahnaz Noroozi and Ziba Eslami. Public key authenticated encryption with keyword search: revisited. *IET Information Security*, 13(4):336–342, 2019.

[33] Xiangyu Pan and Fagen Li. Public-key authenticated encryption with keyword search achieving both multi-ciphertext and multi-trapdoor indistinguishability. *Journal of Systems Architecture*, 115:102075, 2021.

[34] Baodong Qin, Yu Chen, Qiong Huang, Ximeng Liu, and Dong Zheng. Public-key authenticated encryption with keyword search revisited: Security model and constructions. *Information Sciences*, 516:515–528, 2020.

[35] Baodong Qin, Hui Cui, Xiaokun Zheng, and Dong Zheng. Improved security model for public-key authenticated encryption with keyword search. In *ProvSec*, pages 19–38, 2021.

[36] Ningbin Yang, Quan Zhou, Qiong Huang, and Chunming Tang. Multi-recipient encryption with keyword search without pairing for cloud storage. *Journal of Cloud Computing*, 11:10, 2022.

[37] Lisha Yao, Jian Weng, Anjia Yang, Xiaojian Liang, Zhenghao Wu, Zike Jiang, and Lin Hou. Scalable CCA-secure public-key authenticated encryption with keyword search from ideal lattices in cloud computing. *Information Sciences*, 624:777–795, 2023.

[38] Kai Zhang, Mi Wen, Rongxing Lu, and Kefei Chen. Multi-client sub-linear boolean keyword searching for encrypted cloud storage with owner-enforced authorization. *IEEE Transactions on Dependable and Secure Computing*, 18(6):2875–2887, 2021.