

Research Article

Generic Construction of Fully Anonymous Broadcast Authenticated Encryption with Keyword Search with Adaptive Corruptions

Keita Emura ^{1,2}

¹Kanazawa University, Ishikawa, Japan

²National Institute of Information and Communications Technology (NICT), Tokyo, Japan

Correspondence should be addressed to Keita Emura; keita.emura@gmail.com

Received 3 June 2023; Revised 22 September 2023; Accepted 3 October 2023; Published 7 November 2023

Academic Editor: Hyun-A Park

Copyright © 2023 Keita Emura. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As a multireceiver 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 multireceiver setting. The proposed construction is an extension of the generic construction proposed by Libert et al. (PKC 2012) for the fully 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 multireceiver setting and can be employed as a building block of the proposed generic construction.

1. Introduction

Public key authenticated encryption with keyword search (PAEKS) [1–12] has been proposed as an extension of public key encryption with keyword search (PEKS) [13]. In PAEKS, a sender's secret key is required for encryption. Because of the restriction of the rights of encryption, a keyword guessing attack (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.) is prevented. PAEKS requires that no information about the keyword is leaked from both the ciphertexts and trapdoors. Some recent research progresses are given, e.g., [14–18].

Broadcast authenticated encryption with keyword search (BAEKS) [19] was proposed by Liu et al. [19] as a multireceiver variant of PAEKS. Unlike other multireceiver variants of P(A)EKS [20–28] (Attrapadung et al. [21] introduced

broadcast encryption with keyword search (BEKS), whose security is defined as a selective manner. Chatterjee and Mukherjee [22] proposed a BEKS scheme that 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. [29] on [30] or on [31] also provides pairing-based BEKS constructions. Note that Chatterjee and Mukherjee [22] called a BEKS scheme anonymous if the challenge ciphertext hides the 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.) BAEKS focuses on receiver anonymity, where no information about the receiver is leaked from ciphertexts, which is reminiscent of the anonymous broadcast encryption [32–37]. 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's public key 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 (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 [19], we only focus on the searching phase in this paper.) 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 . As an application of BAEKS, Liu et al. [19] introduced a privacy-preserving payer-view system where a service provider stored various videos in cloud storage, and videos are labeled by the content type or the genre. Then, viewers would like to search by indicating a type or genre such as movie and comedy. BAEKS is effective when videos are encrypted.

Liu et al. [19] 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.
 - (i) If a PAEKS scheme needs to provide 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.
 - (ii) We note that Liu et al. [19] construction defines when the test algorithm outputs 0.
- (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.

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.

1.1. 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. (We do not consider chosen-ciphertext attack (CCA) in this paper, and we omit one-time signatures from the construction hereafter.) Libert et al. [37] 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 [38, 39], i.e., the decryption algorithm outputs the error symbol \perp when a nonappropriate 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.

1.2. Toward 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's 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 multisender setting defined by Emura [5], 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 multireceiver setting.

1.3. 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 multireceiver setting. The proposed construction is an extension of the generic construction of the anonymous broadcast encryption [37] and provides adaptive corruptions. We also demonstrate that the Qin et al. [11] PAEKS scheme provides consistency in a multireceiver setting and ciphertext anonymity. A comparison of our instantiation with the Lin et al. [19] BAEKS scheme is presented in Table 1. The number of test attempts and the ciphertext size are the same as those reported by Liu et al. [19], although the proposed construction provides a higher security level in terms of ciphertext anonymity and adaptive corruptions. We note that the Qin et al. [11] PAEKS scheme does not provide trapdoor anonymity. Consequently, our construction does not provide trapdoor anonymity. However, we argue that trapdoor

TABLE 1: Comparison between our instantiation from the Qin et al. [11] PAEKS scheme and the Liu et al. [19] BAEKS scheme.

Scheme	CT Size	Test Attempts	Consistency	CT Anon.	TD Anon.	Corruption
Liu et al. [19]	$O(N)$	$O(N)$	Not defined	Restricted	Yes	No
Ours (Section 4) + [11]	$O(N)$	$O(N)$	Defined	Full	No	Adaptive

Let S be a set of receivers specified in the encryption algorithm and $N = |S|$. CT and TD stand for ciphertext and trapdoor, respectively. We emphasize that our generic construction provides trapdoor anonymity if the underlying PAEKS scheme provides trapdoor anonymity.

anonymity is not necessary, at least for the setting considered by Liu et al. [19]. 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.

2. Definitions of PAEKS in the Multireceiver Setting

In this section, we introduce the definitions of PAEKS in the multireceiver setting.

Definition 1. (Syntax of PAEKS). A PAEKS scheme PAEKS consists of the following six algorithms (PAEKS.Setup, PAEKS.KG_R, PAEKS.KG_S, PAEKS.Enc, PAEKS.Trapdoor, PAEKS.Test) defined as follows:

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} .

PAEKS.KG_R: The receiver key generation algorithm takes pp as input, and outputs a public key pk_R and secret key sk_R .

PAEKS.KG_S: The sender key generation algorithm takes pp as input, and outputs a public key pk_S and secret key sk_S .

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

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

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

Definition 2. (Correctness). For any security parameter λ , any common parameter $\text{pp} \leftarrow \text{PAEKS.Setup}(1^\lambda)$, any key pairs $(\text{pk}_R, \text{sk}_R) \leftarrow \text{PAEKS.KG}_R(\text{pp})$ and $(\text{pk}_S, \text{sk}_S) \leftarrow \text{PAEKS.KG}_S(\text{pp})$, and any keyword $kw \in \mathcal{KS}$, let $\text{ct}_{\text{PAEKS}} \leftarrow \text{PAEKS.Enc}(\text{pk}_R, \text{pk}_S, \text{sk}_S, kw)$ and $\text{td}_{R, kw} \leftarrow \text{PAEKS.Trapdoor}(\text{pk}_R, \text{pk}_S, \text{sk}_R, kw)$. Then $\Pr[\text{PAEKS.Test}(\text{ct}_{\text{PAEKS}}, \text{td}_{R, kw}) = 1] = 1 - \text{negl}(\lambda)$ holds.

Next, we define computational consistency in the multireceiver 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 the study of Emura [5], the following definition can be extended to consider the multi-sender setting if necessary.

Definition 3. (Computational consistency for multireceivers). For all probabilistic polynomial-time (PPT) adversaries \mathcal{A} , we define the following experiment.

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

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

$$\text{Adv}_{\text{PAEKS}, \mathcal{A}}^{\text{consist}}(\lambda) := \Pr \left[\text{Exp}_{\text{PAEKS}, \mathcal{A}}^{\text{consist}}(\lambda) = 1 \right], \quad (2)$$

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. Our definition also captures ciphertext anonymity simultaneously, i.e., $\text{pk}_{\mathbb{R}[b]}^*$ is used for generating the challenge ciphertext. If we explicitly mention the IND-CKA security in the nonanonymous setting, then $(\text{pk}_{\mathbb{R}[0]}^*, \text{sk}_{\mathbb{R}[0]}^*) = (\text{pk}_{\mathbb{R}[1]}^*, \text{sk}_{\mathbb{R}[1]}^*)$ in the following experiment.

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

$$\begin{aligned} & \text{Exp}_{\text{PAEKS}, \mathcal{A}}^{\text{IND-CKA}}(\lambda) : \\ & \text{pp} \leftarrow \text{PAEKS.Setup}(1^\lambda) \\ & (\text{pk}_{\mathbb{R}[0]}^*, \text{sk}_{\mathbb{R}[0]}^*) \leftarrow \text{PAEKS.KG}_{\mathbb{R}}(\text{pp}); (\text{pk}_{\mathbb{R}[1]}^*, \text{sk}_{\mathbb{R}[1]}^*) \leftarrow \text{PAEKS.KG}_{\mathbb{R}}(\text{pp}) \\ & (\text{pk}_{\mathbb{S}}, \text{sk}_{\mathbb{S}}) \leftarrow \text{PAEKS.KG}_{\mathbb{S}}(\text{pp}) \\ & (kw_0^*, kw_1^*, \text{state}) \leftarrow \mathcal{A}^{\mathcal{O}} \left(\text{pp}, \text{pk}_{\mathbb{R}[0]}^*, \text{pk}_{\mathbb{R}[1]}^*, \text{pk}_{\mathbb{S}} \right) \text{ s.t. } kw_0^*, kw_1^* \in \mathcal{KS} \wedge kw_0^* \neq kw_1^* \\ & b \xleftarrow{\$} \{0, 1\}; \text{ct}_{\text{PAEKS}}^* \leftarrow \text{PAEKS.Enc} \left(\text{pk}_{\mathbb{R}[b]}^*, \text{pk}_{\mathbb{S}}, \text{sk}_{\mathbb{S}}, kw_b^* \right) \\ & b' \leftarrow \mathcal{A}^{\mathcal{O}} \left(\text{state}, \text{ct}_{\text{PAEKS}}^* \right) \\ & \text{If } b = b' \text{ then output 1, and 0 otherwise.} \end{aligned} \quad (3)$$

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 $\text{PAEKS.Enc}(\text{pk}_{\mathbb{R}[i]}^*, \text{pk}_{\mathbb{S}}, \text{sk}_{\mathbb{S}}, kw)$. 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 $\text{PAEKS.Trapdoor}(\text{pk}_{\mathbb{R}[i]}^*, \text{pk}_{\mathbb{S}}, \text{sk}_{\mathbb{R}[i]}, kw')$. 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

$$\text{Adv}_{\text{PAEKS}, \mathcal{A}}^{\text{IND-CKA}}(\lambda) := \left| \Pr \left[\text{Exp}_{\text{PAEKS}, \mathcal{A}}^{\text{IND-CKA}}(\lambda) = 1 \right] - 1/2 \right|, \quad (4)$$

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. Our definition also captures trapdoor anonymity simultaneously, i.e., $\text{pk}_{\mathbb{R}[b]}^*$ is used for generating the challenge trapdoor.

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

$$\begin{aligned} & \text{Exp}_{\text{PAEKS}, \mathcal{A}}^{\text{IND-IKGA}}(\lambda) : \\ & \text{pp} \leftarrow \text{PAEKS.Setup}(1^\lambda) \\ & (\text{pk}_{\mathbb{R}[0]}^*, \text{sk}_{\mathbb{R}[0]}^*) \leftarrow \text{PAEKS.KG}_{\mathbb{R}}(\text{pp}); (\text{pk}_{\mathbb{R}[1]}^*, \text{sk}_{\mathbb{R}[1]}^*) \leftarrow \text{PAEKS.KG}_{\mathbb{R}}(\text{pp}) \\ & (\text{pk}_{\mathbb{S}}, \text{sk}_{\mathbb{S}}) \leftarrow \text{PAEKS.KG}_{\mathbb{S}}(\text{pp}) \\ & (kw_0^*, kw_1^*, \text{state}) \leftarrow \mathcal{A}^{\mathcal{O}} \left(\text{pp}, \text{pk}_{\mathbb{R}[0]}^*, \text{pk}_{\mathbb{R}[1]}^*, \text{pk}_{\mathbb{S}} \right) \text{ s.t. } kw_0^*, kw_1^* \in \mathcal{KS} \wedge kw_0^* \neq kw_1^* \\ & b \xleftarrow{\$} \{0, 1\}; \text{td}_{\mathbb{S}, kw_b^*}^* \leftarrow \text{PAEKS.Trapdoor} \left(\text{pk}_{\mathbb{R}[b]}^*, \text{pk}_{\mathbb{S}}, \text{sk}_{\mathbb{R}[b]}^*, kw_b^* \right) \\ & b' \leftarrow \mathcal{A}^{\mathcal{O}} \left(\text{state}, \text{td}_{\mathbb{S}, kw_b^*}^* \right) \\ & \text{If } b = b' \text{ then output 1, and 0 otherwise.} \end{aligned} \quad (5)$$

Here, $\mathcal{O} := \{\mathcal{O}_C(\cdot, \cdot), \mathcal{O}_T(\text{pk}_R, \cdot, \text{sk}_R, \cdot)\}$. \mathcal{O}_C takes $kw \in \mathcal{KS}$ and $i \in \{0, 1\}$ as input, and returns the result of $\text{PAEKS.Enc}(\text{pk}_{R[i]}, \text{pk}_S, \text{sk}_S, kw)$. 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 $\text{PAEKS.Trapdoor}(\text{pk}_{R[i]}, \text{pk}_S, \text{sk}_{R[i]}, kw')$. 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

$$\text{Adv}_{\text{PAEKS}, \mathcal{A}}^{\text{IND-IKGA}}(\lambda) := \left| \Pr \left[\text{Exp}_{\text{PAEKS}, \mathcal{A}}^{\text{IND-IKGA}}(\lambda) = 1 \right] - 1/2 \right|, \quad (6)$$

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 by Liu et al. [19] but modify them to capture adaptive corruptions.

Definition 6. (Syntax of BAEKS). A BAEKS scheme BAEKS consists of the following six algorithms (BAEKS.Setup, BAEKS.KG_R, BAEKS.KG_S, BAEKS.Enc, BAEKS.Trapdoor, BAEKS.Test) defined as follows:

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} .

BAEKS.KG_R: The receiver key generation algorithm takes pp as input, and outputs a public key pk_R and secret key sk_R .

BAEKS.KG_S: The sender key generation algorithm takes pp as input, and outputs a public key pk_S and secret key sk_S .

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

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

BAEKS.Test: The test algorithm takes ct_{BAEKS} and $\text{td}_{R, kw'}$ 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's secret key and the receiver's 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 multireceiver setting) of the underlying PAEKS scheme.

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

$$\begin{aligned} & \text{Exp}_{\text{BAEKS}, \mathcal{A}}^{\text{correct}}(\lambda): \\ & \text{pp} \leftarrow \text{BAEKS.Setup}(1^\lambda, N_{\max}) \\ & \text{For } i \in [1, N_{\max}], (\text{pk}_{R[i]}, \text{sk}_{R[i]}) \leftarrow \text{BAEKS.KG}_R(\text{pp}) \\ & (\text{pk}_S, \text{sk}_S) \leftarrow \text{BAEKS.KG}_S(\text{pp}) \\ & (kw, S, \text{pk}_R) \leftarrow \mathcal{A}(\text{pp}, \{\text{pk}_{R[i]}\}_{i \in [1, N_{\max}]}, \text{pk}_S) \text{ s.t. } kw \in \mathcal{KS} \wedge S \subseteq \{\text{pk}_{R[1]}, \dots, \text{pk}_{R[N_{\max}]}\} \wedge \text{pk}_R \in S \\ & \text{ct}_{\text{BAEKS}} \leftarrow \text{BAEKS.Enc}(\text{pp}, S, \text{pk}_S, \text{sk}_S, kw); \text{td}_{R, kw} \leftarrow \text{BAEKS.Trapdoor}(\text{pk}_R, \text{pk}_S, \text{sk}_R, kw) \\ & \text{If } \text{BAEKS.Test}(\text{ct}_{\text{BAEKS}}, \text{td}_{R, kw}) = 1, \text{ then output 1, and 0 otherwise.} \end{aligned} \quad (7)$$

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

$$\text{Adv}_{\text{BAEKS}, \mathcal{A}}^{\text{correct}}(\lambda) := \Pr \left[\text{Exp}_{\text{BAEKS}, \mathcal{A}}^{\text{correct}}(\lambda) = 1 \right], \quad (8)$$

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 $\text{pk}_R \notin S$, then $\text{BAEKS.Test}(\text{ct}_{\text{BAEKS}}, \text{td}_{R, kw'}) = 0$ holds even if $kw = kw'$, where $\text{ct}_{\text{BAEKS}} \leftarrow \text{BAEKS.Enc}(\text{pp}, S, \text{pk}_S, \text{sk}_S, kw)$ and $\text{td}_{R, kw'} \leftarrow \text{BAEKS.Trapdoor}(\text{pk}_R, \text{pk}_S, \text{sk}_R, kw')$. The following definition captures this case by the condition $(kw \neq kw' \vee \text{pk}_R \notin S)$.

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

$$\begin{aligned}
& \text{Exp}_{\text{BAEKS}, \mathcal{A}}^{\text{consist}}(\lambda): \\
& \text{pp} \leftarrow \text{BAEKS.Setup}(1^\lambda, N_{\max}) \\
& \text{For } i \in [1, N_{\max}], (\text{pk}_{R[i]}, \text{sk}_{R[i]}) \leftarrow \text{BAEKS.KG}_R(\text{pp}) \\
& (\text{pk}_S, \text{sk}_S) \leftarrow \text{BAEKS.KG}_S(\text{pp}) \\
& (kw, kw', S, \text{pk}_R) \leftarrow \mathcal{A}(\text{pp}, \{\text{pk}_{R[i]}\}_{i \in [1, N_{\max}]}, \text{pk}_S) \\
& \text{s.t. } kw, kw' \in \mathcal{KS} \wedge S \subseteq \{\text{pk}_{R[1]}, \dots, \text{pk}_{R[N_{\max}]}\} \wedge (kw \neq kw' \vee \text{pk}_R \notin S) \\
& \text{ct}_{\text{BAEKS}} \leftarrow \text{BAEKS.Enc}(\text{pp}, S, \text{pk}_S, \text{sk}_S, kw) \\
& \text{td}_{R, kw'} \leftarrow \text{BAEKS.Trapdoor}(\text{pk}_R, \text{pk}_S, \text{sk}_R, kw') \\
& \text{If } \text{BAEKS.Test}(\text{ct}_{\text{BAEKS}}, \text{td}_{R, kw'}) = 1, \text{ then output 1, and 0 otherwise.}
\end{aligned} \tag{9}$$

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

$$\text{Adv}_{\text{BAEKS}, \mathcal{A}}^{\text{consist}}(\lambda) := \Pr \left[\text{Exp}_{\text{BAEKS}, \mathcal{A}}^{\text{consist}}(\lambda) = 1 \right], \tag{10}$$

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. Our defi-

inition also captures 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 $\text{sk}_{R[i]}$. If $\text{pk}_{R[i]} \in S_0^* \cap S_1^*$, then $kw_0^* = kw_1^*$ is required to hold. Adversaries \mathcal{A} are also allowed to obtain trapdoors generated by $\text{sk}_{R[i]}$. Similarly, if $\text{pk}_{R[i]} \in S_0^* \cap S_1^*$, then $kw_0^* = kw_1^*$ is required to hold.

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

$$\begin{aligned}
& \text{Exp}_{\text{BAEKS}, \mathcal{A}}^{\text{IND-CKA}}(\lambda): \\
& \text{pp} \leftarrow \text{BAEKS.Setup}(1^\lambda, N_{\max}) \\
& \text{For } i \in [1, N_{\max}], (\text{pk}_{R[i]}, \text{sk}_{R[i]}) \leftarrow \text{BAEKS.KG}_R(\text{pp}) \\
& (\text{pk}_S, \text{sk}_S) \leftarrow \text{BAEKS.KG}_S(\text{pp}) \\
& (kw_0^*, kw_1^*, S_0^*, S_1^*, \text{state}) \leftarrow \mathcal{A}^\theta(\text{pp}, \{\text{pk}_{R[i]}\}_{i \in [1, N_{\max}]}, \text{pk}_S) \\
& \text{s.t. } kw_0^*, kw_1^* \in \mathcal{KS} \wedge S_0^*, S_1^* \subseteq \{\text{pk}_{R[1]}, \dots, \text{pk}_{R[N_{\max}]}\} \wedge |S_0^*| = |S_1^*| \\
& b \xleftarrow{\$} \{0, 1\}; \text{ct}_{\text{BAEKS}}^* \leftarrow \text{BAEKS.Enc}(\text{pp}, S_b^*, \text{pk}_S, \text{sk}_S, kw_b^*) \\
& b' \leftarrow \mathcal{A}^\theta(\text{state}, \text{ct}_{\text{BAEKS}}^*) \\
& \text{If } b = b' \text{ then output 1, and 0 otherwise.}
\end{aligned} \tag{11}$$

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

$$\text{Adv}_{\text{BAEKS}, \mathcal{A}}^{\text{IND-CKA}}(\lambda) := \left| \Pr \left[\text{Exp}_{\text{BAEKS}, \mathcal{A}}^{\text{IND-CKA}}(\lambda) = 1 \right] - 1/2 \right|, \tag{12}$$

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. Our definition also captures trapdoor anonymity simultaneously, i.e., $(\text{pk}_{R[b]}, \text{sk}_{R[b]})$ are used for generating the challenge trapdoor.

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

$$\begin{aligned}
& \text{Exp}_{\text{BAEKS}, \mathcal{A}}^{\text{IND-IKGA}}(\lambda) : \\
& \text{pp} \leftarrow \text{BAEKS.Setup}(1^\lambda, N_{\max}) \\
& \text{For } i \in [1, N_{\max}], (\text{pk}_{\text{R}[i]}, \text{sk}_{\text{R}[i]}) \leftarrow \text{BAEKS.KG}_{\text{R}}(\text{pp}) \\
& (\text{pk}_{\text{S}}, \text{sk}_{\text{S}}) \leftarrow \text{BAEKS.KG}_{\text{S}}(\text{pp}) \\
& (kw_0^*, kw_1^*, \text{pk}_{\text{R}[0]}^*, \text{pk}_{\text{R}[1]}^*, \text{state}) \leftarrow \mathcal{A}^{\mathcal{O}}(\text{pp}, \{\text{pk}_{\text{R}[i]}\}_{i \in [1, N_{\max}]}, \text{pk}_{\text{S}}) \\
& \text{s.t. } kw_0^*, kw_1^* \in \mathcal{KS} \wedge kw_0^* \neq kw_1^* \wedge \text{pk}_{\text{R}[0]}^*, \text{pk}_{\text{R}[1]}^* \in \{\text{pk}_{\text{R}[i]}\}_{i \in [1, N_{\max}]} \\
& b \xleftarrow{\$} \{0, 1\}; \text{td}_{\text{R}[b], kw_b^*}^* \leftarrow \text{BAEKS.Trapdoor}(\text{pk}_{\text{R}[b]}^*, \text{pk}_{\text{S}}, \text{sk}_{\text{R}[b]}^*, kw_b^*) \\
& b' \leftarrow \mathcal{A}^{\mathcal{O}}(\text{state}, \text{td}_{\text{R}[b], kw_b^*}^*) \\
& \text{If } b = b' \text{ then output 1, and 0 otherwise.}
\end{aligned} \tag{13}$$

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

$$\text{Adv}_{\text{BAEKS}, \mathcal{A}}^{\text{IND-IKGA}}(\lambda) := \left| \Pr \left[\text{Exp}_{\text{BAEKS}, \mathcal{A}}^{\text{IND-IKGA}}(\lambda) = 1 \right] - 1/2 \right|, \tag{14}$$

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 $\text{PAEKS} = (\text{PAEKS.Setup}, \text{PAEKS.KG}_{\text{R}}, \text{PAEKS.KG}_{\text{S}}, \text{PAEKS.Enc}, \text{PAEKS.Trapdoor}, \text{PAEKS.Test})$ be a PAEKS scheme. Let $\pi: \{1, \dots, N\} \rightarrow \{1, \dots, N\}$ be a random permutation for any $N \leq N_{\max}$. Intuitively, a BAEKS ciphertext is a set of PAEKS ciphertexts with each public key $\text{pk}_{\text{R}[i]}$ and the same keyword kw . Due to consistency in the multireceiver setting, the test algorithm of the underlying PAEKS scheme outputs 0 for a ciphertext encrypted by $\text{pk}_{\text{R}[i]}$ and a trapdoor generated by $\text{pk}_{\text{R}[j]}$ and $i \neq j$, even if kw is associated to the trapdoor. That is consistency in the multireceiver setting acts as robustness in the generic construction of anonymous broadcast encryption. Moreover, a BAEKS ciphertext $(\text{ct}_{\text{PAEKS}1}, \dots, \text{ct}_{\text{PAEKS}N})$ is randomly sorted by a random permutation π such that

$(\text{ct}_{\text{PAEKS}\pi(1)}, \dots, \text{ct}_{\text{PAEKS}\pi(N)})$. Thus, no information about the receiver is revealed, at least from the order of ciphertexts. The construction of a BAEKS scheme BAEKS from PAEKS is described below.

4.1. The Proposed Generic Construction

$$\begin{aligned}
& \text{BAEKS.Setup}(\lambda, N_{\max}): \text{Run } \text{pp}' \leftarrow \text{PAEKS.Setup}(1^\lambda) \text{ and output } \text{pp} = (\text{pp}', N_{\max}). \\
& \text{BAEKS.KG}_{\text{R}}(\text{pp}): \text{Parse } \text{pp} = (\text{pp}', N_{\max}). \text{Run } (\text{pk}_{\text{R}}, \text{sk}_{\text{R}}) \leftarrow \text{PAEKS.KG}_{\text{R}}(\text{pp}') \text{ and output } (\text{pk}_{\text{R}}, \text{sk}_{\text{R}}). \\
& \text{BAEKS.KG}_{\text{S}}(\text{pp}): \text{Parse } \text{pp} = (\text{pp}', N_{\max}). \text{Run } (\text{pk}_{\text{S}}, \text{sk}_{\text{S}}) \leftarrow \text{PAEKS.KG}_{\text{S}}(\text{pp}') \text{ and output } (\text{pk}_{\text{S}}, \text{sk}_{\text{S}}). \\
& \text{BAEKS.Enc}(\text{pp}, S, \text{pk}_{\text{S}}, \text{sk}_{\text{S}}, kw): \text{Parse } \text{pp} = (\text{pp}', N_{\max}). \text{Without loss of generality, we denote } S = \{\text{pk}_{\text{R}[1]}, \dots, \text{pk}_{\text{R}[N]}\}. \text{For all } i \in [1, N_{\max}], \text{run } \text{ct}_{\text{PAEKS}i} \leftarrow \text{PAEKS.Enc}(\text{pk}_{\text{R}[i]}, \text{pk}_{\text{S}}, \text{sk}_{\text{S}}, kw). \\
& \text{Output } \text{ct}_{\text{BAEKS}} = \{\text{ct}_{\text{PAEKS}\pi(i)}\}_{i \in [1, N]}. \\
& \text{BAEKS.Trapdoor}(\text{pk}_{\text{R}}, \text{pk}_{\text{S}}, \text{sk}_{\text{R}}, kw'): \text{Run } \text{td}_{\text{R}, kw'} \leftarrow \text{PAEKS.Trapdoor}(\text{pk}_{\text{R}}, \text{pk}_{\text{S}}, \text{sk}_{\text{R}}, kw) \text{ and output } \text{td}_{\text{R}, kw'}. \\
& \text{BAEKS.Test}(\text{ct}_{\text{BAEKS}}, \text{td}_{\text{R}, kw'}): \text{Parse } \text{ct}_{\text{BAEKS}} = \{\text{ct}_{\text{PAEKS}i}\}_{i \in [1, N]}. \text{Output 1 if there exists } i \in [1, N] \text{ such that } \text{PAEKS.Test}(\text{ct}_{\text{PAEKS}i}, \text{td}_{\text{R}, kw'}) = 1, \text{ and 0 otherwise.}
\end{aligned}$$

Because of consistency in the multireceiver setting of PAEKS, $\text{PAEKS.Test}(\text{ct}_{\text{PAEKS}i}, \text{td}_{\text{R}[j], kw'}) = 0$ for $\text{ct}_{\text{PAEKS}i} \leftarrow \text{PAEKS.Enc}(\text{pk}_{\text{R}[i]}, \text{pk}_{\text{S}}, \text{sk}_{\text{S}}, kw)$ and $\text{td}_{\text{R}[j], kw'} \leftarrow \text{PAEKS.Trapdoor}(\text{pk}_{\text{R}[j]}, \text{pk}_{\text{S}}, \text{sk}_{\text{R}[j]}, kw')$ if $(kw, i) \neq (kw', j)$. Thus, the proposed construction is correct.

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

5. Security Analysis

In this section, we prove the following theorems. We note that Libert et al. [37] 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-CCA/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 $|S_0^*| = |S_1^*| = N^* \leq N_{\max}$ and $\ell = |S_0^* \cap S_1^*|$. \square

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 $\tilde{S}_0^* = \{\theta_1, \dots, \theta_\ell, \theta_{\ell+1}, \dots, \theta_{N^*}\}$ and $\tilde{S}_1^* = \{\rho_1, \dots, \rho_\ell, \rho_{\ell+1}, \dots, \rho_{N^*}\}$, 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 $\text{ct}_{\text{BAEKS}}^*$ is generated as follows:

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

Then, $\text{ct}_{\text{BAEKS}}^* = \{\text{ct}_{\text{PAEKS}\pi(i)}^*\}_{i \in [1, N^*]}$. 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 $\tilde{S}_0^* = \{\theta_1, \dots, \theta_\ell, \theta_{\ell+1}, \dots, \theta_{N^*}\}$ and $\tilde{S}_1^* = \{\rho_1, \dots, \rho_\ell, \rho_{\ell+1}, \dots, \rho_{N^*}\}$, 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 $\text{ct}_{\text{BAEKS}}^*$ is generated as follows:

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

Then, $\text{ct}_{\text{BAEKS}}^* = \{\text{ct}_{\text{PAEKS}\pi(i)}^*\}_{i \in [1, N^*]}$. 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*

$$|\Pr[E_k] - \Pr[E_{k-1}]| \leq N_{\max} \cdot \text{Adv}_{\text{PAEKS}, \mathcal{B}}^{\text{IND-CKA}}(\lambda). \quad (15)$$

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 $\mathcal{O}_{\text{Ext}}(\text{pk}_{\text{R}[i]})$ such that $\text{pk}_{\text{R}[i]} \in S_0^* \cap S_1^*$, 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 $\mathcal{O}_{\text{Ext}}(\text{pk}_{\text{R}[i]})$ for $\text{pk}_{\text{R}[i]} \in S_0^* \cap S_1^*$.

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

- (i) When \mathcal{A} issues $\mathcal{O}_{\text{C}}(kw, S)$ where $|S| = N$, if $\text{pk}_{\text{R}}^* \in S$, then \mathcal{B} issues $\mathcal{O}_{\text{C}}(kw, 0)$ of the underlying PAEKS scheme, obtains $\text{ct}_{\text{PAEKS}} \leftarrow \text{PAEKS.Enc}(\text{pk}_{\text{R}}^*, \text{pk}_{\text{S}}, \text{sk}_{\text{S}}, kw)$, and sets $\text{ct}_{\text{PAEKS}_{i^*}} = \text{ct}_{\text{PAEKS}}$. \mathcal{B} generates other PAEKS ciphertexts using $\text{sk}_{\text{R}[i]}$. \mathcal{B} returns $\text{ct}_{\text{BAEKS}} = \{\text{ct}_{\text{PAEKS}\pi(i)}\}_{i \in [1, N]}$ to \mathcal{A} .
- (ii) When \mathcal{A} issues $\mathcal{O}_{\text{T}}(kw', \text{pk}_{\text{R}[i]})$, if $i = i^*$, then \mathcal{B} issues $\mathcal{O}_{\text{T}}(kw', 0)$ of the underlying PAEKS scheme, obtains $\text{td}_{\text{R}, kw'} \leftarrow \text{PAEKS.Trapdoor}(\text{pk}_{\text{R}}^*, \text{pk}_{\text{S}}, \text{sk}_{\text{R}}^*, kw')$, and sends $\text{td}_{\text{R}, kw'}$ to \mathcal{A} . If $i \neq i^*$, then \mathcal{B} responds the query using $\text{sk}_{\text{R}[i]}$.
- (iii) When \mathcal{A} issues $\mathcal{O}_{\text{Ext}}(\text{pk}_{\text{R}[i]})$ for $i \in [1, N_{\max}] \setminus \{i^*\}$, \mathcal{B} returns $\text{sk}_{\text{R}[i]}$. When \mathcal{A} issues $\mathcal{O}_{\text{Ext}}(\text{pk}_{\text{R}[i^*]})$, \mathcal{B} aborts.

In the challenge phase, \mathcal{A} declares $(kw_0^*, kw_1^*, S_0^*, S_1^*)$. \mathcal{B} reorders indices of S_0^* and S_1^* such that $\tilde{S}_0^* = \{\theta_1, \dots, \theta_\ell, \theta_{\ell+1}, \dots, \theta_{N^*}\}$ and $\tilde{S}_1^* = \{\rho_1, \dots, \rho_\ell, \rho_{\ell+1}, \dots, \rho_{N^*}\}$ 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 $1/N_{\max}$ since the choice of i^* is completely independent of \mathcal{A} 's view. We remark that if $\theta_k = i^*$, then $\text{pk}_{\text{R}[i^*]} = \text{pk}_{\text{R}}^* \in S_0^* \cap S_1^*$. Thus, \mathcal{A} does not issue $\mathcal{O}_{\text{Ext}}(\text{pk}_{\text{R}[i^*]})$ as mentioned above. \mathcal{B} sends (kw_0^*, kw_1^*) to \mathcal{C} as the challenge keywords. \mathcal{C} sends $\text{ct}_{\text{PAEKS}}^* \leftarrow \text{PAEKS.Enc}(\text{pk}_{\text{R}}^*, \text{pk}_{\text{S}}, \text{sk}_{\text{S}}, kw_b^*)$ to \mathcal{B} for some internally flipped random bit $b \xleftarrow{\$} \{0, 1\}$. The BAEKS challenge ciphertext $\text{ct}_{\text{BAEKS}}^* = \{\text{ct}_{\text{PAEKS}\pi(i)}^*\}_{i \in [1, N^*]}$ is generated as follows:

- (i) For $j \in [1, k-1]$, \mathcal{B} issues $\mathcal{O}_C(kw_1^*, \text{pk}_{R[\theta_j]})$. Then \mathcal{C} responds $\text{ct}_{\text{PAEKS}_j} \leftarrow \text{PAEKS.Enc}(\text{pk}_{R[\theta_j]}, \text{pk}_S, \text{sk}_S, kw_1^*)$ to \mathcal{B} .
- (ii) For $j = k$, \mathcal{B} sets $\text{ct}_{\text{PAEKS}_j} = \text{ct}_{\text{PAEKS}}^*$.
- (iii) For $j \in [k+1, N^*]$, \mathcal{B} issues $\mathcal{O}_C(kw_0^*, \text{pk}_{R[\theta_j]})$. Then \mathcal{C} responds $\text{ct}_{\text{PAEKS}_j} \leftarrow \text{PAEKS.Enc}(\text{pk}_{R[\theta_j]}, \text{pk}_S, \text{sk}_S, kw_0^*)$ 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. \square

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*

$$|\Pr[E_{k'}] - \Pr[E_{k'-1}]| \leq N_{\max}^2 \cdot \text{Adv}_{\text{PAEKS}, \mathcal{B}}^{\text{IND-CKA}}(\lambda) \quad (16)$$

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 $(\text{pp}', \text{pk}_{R[0]}^*, \text{pk}_{R[1]}^*, \text{pk}_S)$ from \mathcal{C} . \mathcal{B} picks two distinct indices $i_0^*, i_1^* \xleftarrow{\$} \{1, N_{\max}\}$ and sets $\text{pk}_{R[i_0^*]} = \text{pk}_{R[0]}^*$ and $\text{pk}_{R[i_1^*]} = \text{pk}_{R[1]}^*$. For $i \in [1, N_{\max}] \setminus \{i_0^*, i_1^*\}$, \mathcal{B} runs $(\text{pk}_{R[i]}, \text{sk}_{R[i]}) \leftarrow \text{PAEKS.KG}_R(\text{pp}')$. \mathcal{B} sets $\text{pp} = (\text{pp}', N_{\max})$ and sends $(\text{pp}, \{\text{pk}_{R[i]}\}_{i \in [1, N_{\max}]}, \text{pk}_S)$ to \mathcal{A} .

- (i) When \mathcal{A} issues $\mathcal{O}_C(kw, S)$ where $|S| = N$, if $\text{pk}_{R[i_0^*]} \in S$, then \mathcal{B} issues $\mathcal{O}_C(kw, 0)$ of the underlying PAEKS scheme, obtains $\text{ct}_{\text{PAEKS}} \leftarrow \text{PAEKS.Enc}(\text{pk}_{R[0]}^*, \text{pk}_S, \text{sk}_S, kw)$, and sets $\text{ct}_{\text{PAEKS}_{i_0^*}} = \text{ct}_{\text{PAEKS}}$. If $\text{pk}_{R[i_1^*]} \in S$, then \mathcal{B} issues $\mathcal{O}_C(kw, 1)$ of the underlying PAEKS scheme, obtains $\text{ct}_{\text{PAEKS}} \leftarrow \text{PAEKS.Enc}(\text{pk}_{R[1]}^*, \text{pk}_S, \text{sk}_S, kw)$, and sets $\text{ct}_{\text{PAEKS}_{i_1^*}} = \text{ct}_{\text{PAEKS}}$. \mathcal{B} generates other PAEKS ciphertexts using $\text{sk}_{R[i]}$. \mathcal{B} returns $\text{ct}_{\text{BAEKS}} = \{\text{ct}_{\text{PAEKS}_{\pi(i)}}\}_{i \in [1, N]}$ to \mathcal{A} .
- (ii) When \mathcal{A} issues $\mathcal{O}_T(kw', \text{pk}_{R[i]})$, if $i = i_0^*$, then \mathcal{B} issues $\mathcal{O}_T(kw', 0)$ of the underlying PAEKS scheme, obtains $\text{td}_{R, kw'} \leftarrow \text{PAEKS.Trapdoor}(\text{pk}_{R[0]}^*, \text{pk}_S, \text{sk}_{R[0]}^*, kw')$, and sends $\text{td}_{R, kw'}$ to \mathcal{A} . If $i = i_1^*$, then \mathcal{B} issues $\mathcal{O}_T(kw', 1)$ of the underlying PAEKS scheme, obtains $\text{td}_{R, kw'} \leftarrow \text{PAEKS.Trapdoor}(\text{pk}_{R[1]}^*, \text{pk}_S, \text{sk}_{R[1]}^*, kw')$, and

sends $\text{td}_{R, kw'}$ to \mathcal{A} . If $i \notin \{i_0^*, i_1^*\}$, then \mathcal{B} responds the query using $\text{sk}_{R[i]}$.

- (iii) When \mathcal{A} issues $\mathcal{O}_{\text{Ext}}(\text{pk}_{R[i]})$ for $i \in [1, N_{\max}] \setminus \{i_0^*, i_1^*\}$, \mathcal{B} returns $\text{sk}_{R[i]}$. When \mathcal{A} issues $\mathcal{O}_{\text{Ext}}(\text{pk}_{R[i]})$ 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} reorders indices of S_0^* and S_1^* such that $\tilde{S}_0^* = \{\theta_1, \dots, \theta_\ell, \theta_{\ell+1}, \dots, \theta_{N^*}\}$ and $\tilde{S}_1^* = \{\rho_1, \dots, \rho_\ell, \rho_{\ell+1}, \dots, \rho_{N^*}\}$, 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 $1/N_{\max}(N_{\max}-1) > 1/N_{\max}^2$ 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 $\text{pk}_{R[i_0^*]}, \text{pk}_{R[i_1^*]} \in S_0^* \Delta S_1^*$ and thus \mathcal{A} does not issue both $\mathcal{O}_{\text{Ext}}(\text{pk}_{R[i_0^*]})$ and $\mathcal{O}_{\text{Ext}}(\text{pk}_{R[i_1^*]})$. \mathcal{B} sends (kw_0^*, kw_1^*) to \mathcal{C} as the challenge keywords. \mathcal{C} sends $\text{ct}_{\text{PAEKS}}^* \leftarrow \text{PAEKS.Enc}(\text{pk}_{R[b]}^*, \text{pk}_S, \text{sk}_S, kw_b^*)$ to \mathcal{B} for some internally flipped random bit $b \xleftarrow{\$} \{0, 1\}$. The BAEKS challenge ciphertext $\text{ct}_{\text{BAEKS}}^* = \{\text{ct}_{\text{PAEKS}_{\pi(i)}}\}_{i \in [1, N^]}$ is generated as follows:

- (i) For $j \in [1, k'-1]$, \mathcal{B} issues $\mathcal{O}_C(kw_1^*, \text{pk}_{R[\theta_j]})$. Then \mathcal{C} responds $\text{ct}_{\text{PAEKS}_j} \leftarrow \text{PAEKS.Enc}(\text{pk}_{R[\theta_j]}, \text{pk}_S, \text{sk}_S, kw_1^*)$ to \mathcal{B} .
- (ii) For $j = k'$, \mathcal{B} sets $\text{ct}_{\text{PAEKS}_j} = \text{ct}_{\text{PAEKS}}^*$.
- (iii) For $j \in [k'+1, N^*]$, \mathcal{B} issues $\mathcal{O}_C(kw_0^*, \text{pk}_{R[\theta_j]})$. Then \mathcal{C} responds $\text{ct}_{\text{PAEKS}_j} \leftarrow \text{PAEKS.Enc}(\text{pk}_{R[\theta_j]}, \text{pk}_S, \text{sk}_S, kw_0^*)$ 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.

From Lemma 1 and Lemma 2, we have $|\Pr[E_0] - \Pr[E_{N^*}]| \leq \ell \cdot N_{\max} \cdot \text{Adv}_{\text{PAEKS}, \mathcal{B}}^{\text{IND-CKA}}(\lambda) + (N^* - \ell) \cdot N_{\max}^2 \cdot \text{Adv}_{\text{PAEKS}, \mathcal{B}}^{\text{IND-CKA}}(\lambda) \leq N_{\max}^3 \cdot \text{Adv}_{\text{PAEKS}, \mathcal{B}}^{\text{IND-CKA}}(\lambda)$. 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 $(\text{pk}_{R[0]}^*, \text{pk}_{R[1]}^*)$, given by the challenger of the IND-IKGA security of PAEKS \mathcal{C} , to $\{\text{pk}_{R[i]}\}_{i \in [1, N_{\max}]}$, and expects that $(\text{pk}_{R[0]}^*, \text{pk}_{R[1]}^*)$ 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. [11] PAEKS

In this section, we briefly explain that the Qin et al. [11] PAEKS scheme provides consistency in the multireceiver 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. [11] PAEKS scheme is described as follows:

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

PAEKS.KG_R(pp): Choose $x, v \xleftarrow{\$} \mathbb{Z}_p$. Output $\text{pk}_R = (\text{pk}_R^{(1)}, \text{pk}_R^{(2)}) = (g^x, g^v)$ and $\text{sk}_R = (\text{sk}_R^{(1)}, \text{sk}_R^{(2)}) = (x, v)$.

PAEKS.KG_S(pp): Choose $u \xleftarrow{\$} \mathbb{Z}_p$. output $\text{pk}_S = g^u$ and $\text{sk}_S = u$.

PAEKS.Enc($\text{pk}_R, \text{pk}_S, \text{sk}_S, kw$): Parse $\text{pk}_R = (\text{pk}_R^{(1)}, \text{pk}_R^{(2)})$ and $\text{sk}_S = u$. Choose $r \xleftarrow{\$} \mathbb{Z}_p$ and compute $A = g^x$. Compute $\text{DHkey}_{S,R} = (\text{pk}_R^{(2)})^u (= g^{uv})$, $h = H_1(kw \parallel \text{pk}_S \parallel \text{pk}_R \parallel H_3(\text{DHkey}_{S,R}))$, and $B = H_2(e(h^r, \text{pk}_R^{(1)}))$. Output $\text{ct}_{\text{PAEKS}} = (A, B)$.

PAEKS.Trapdoor($\text{pk}_R, \text{pk}_S, \text{sk}_R, kw'$): Parse $\text{pk}_R = (\text{pk}_R^{(1)}, \text{pk}_R^{(2)})$ and $\text{sk}_R = (\text{sk}_R^{(1)}, \text{sk}_R^{(2)})$. Compute $\text{DHkey}_{S,R} = \text{pk}_S^{\text{sk}_R^{(2)}} (= g^{uv})$ and $h' = H_1(kw' \parallel \text{pk}_S \parallel \text{pk}_R \parallel H_3(\text{DHkey}_{S,R}))$. Output $\text{td}_{R,kw'} = (h')^{\text{sk}_R^{(1)}} = (h')^x$.

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

Intuitively, a Diffie–Hellman (DH) key $\text{DHkey}_{S,R} = (\text{pk}_R^{(2)})^{\text{sk}_S} = (\text{pk}_S)^{\text{sk}_R^{(2)}} = g^{uv}$ 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 $h = H_1(kw \parallel \text{pk}_S \parallel \text{pk}_R \parallel H_3(\text{DHkey}_{S,R}))$. 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 $B = H_2(e(h^r, \text{pk}_R^{(1)}))$ 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. [11] proved the IND-CKA security under the bilinear Diffie–Hellman 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 $\text{pk}_{R[0]}$ and $\text{pk}_{R[1]}$, the challenge bit b is hidden from $H_1(kw_b^* \parallel \text{pk}_S \parallel \text{pk}_{R[b]} \parallel H_3(\text{DHkey}_{S,R_b}))$ and the simulation given by Qin et al. [11] still works. The value h' is computed by a keyword to be searched and a DH key, such that $h' = H_1(kw' \parallel \text{pk}_S \parallel \text{pk}_R \parallel H_3(\text{DHkey}_{S,R}))$, and $\text{td}_{R,kw'} = (h')^x$. 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 multireceiver 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 $\text{td}_{R,kw'} = (h')^x$. Formally, Qin et al. [11] introduced the computational oracle Diffie–Hellman (CODH) problem and proved that the scheme provides the IND-IKGA security under the CODH assumption.

However, because $(g, h', \text{pk}_R^{(1)}, \text{td}_{R,kw'}) = (g, h', g^x, (h')^x)$ is a decisional Diffie–Hellman (DDH) tuple, $e(\text{pk}_R^{(1)}, h') = e(g, \text{td}_{R,kw'})$ holds if $\text{td}_{R,kw'}$ is generated by the receiver (whose public key is pk_R). Thus, the Qin et al. [11] PAEKS scheme does not provide trapdoor anonymity. To provide trapdoor anonymity, one may employ type-3 asymmetric pairings; where $e: \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$, 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, $(g, h', \text{pk}_R^{(1)}, \text{td}_{R,kw'})$ must belong to the same group. However, a ciphertext consists of $B = H_2(e(h^r, \text{pk}_R^{(1)}))$, i.e., h and $\text{pk}_R^{(1)}$ belong to different groups, and thus h and h' also belong to different groups. This violates the correctness of the Qin et al. [11] 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 multireceiver setting. Our generic constructions provide adaptive corruptions. We also show that the Qin et al. [11] PAEKS scheme can be employed for instantiating the proposed generic construction.

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. [37] 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. [33] 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. [37] construction provides a linear-size ciphertext, whereas the Fazio et al. [33] construction provides a sublinear-size ciphertext using the subset cover framework [40] at the expense of a weak anonymity level. Although outsider anonymity seems sufficient in some applications, the construction proposed by Fazio et al. [33] cannot be extended to BAEKS directly because Fazio et al. [33] 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 [41], but it does not consider authenticity. Employing the Fazio et al. [33] construction in the BAEKS context is left as a future work.

Though Yao et al. [12] proposed a lattice-based PAEKS scheme, they did not define consistency, and thus it is unclear whether the Yao et al. [12] PAEKS scheme provides consistency in the multireceiver setting. Moreover, they did not define trapdoor privacy (they considered ciphertext privacy that guarantees no information about keywords is revealed from ciphertexts and considered the unforgeability of ciphertexts and trapdoors). Thus, we do not consider the Yao et al. [12] scheme as a building block of the proposed generic construction. Cheng and Meng [3] 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.

Data Availability

No underlying data were collected or produced in this study.

Disclosure

The main part of the study was done when the author was with the National Institute of Information and Communications Technology (NICT), Japan. The preprint version has been posted [42].

Conflicts of Interest

The author declares that there is no conflicts of interest.

Acknowledgments

This work was supported by JSPS KAKENHI (grant number JP21K11897).

References

- [1] M. Calderini, R. Longo, M. Sala, and I. Villa, "Searchable encryption with randomized ciphertext and randomized keyword search," *IACR Cryptology ePrint Archive*, Article ID 945, 2022.
- [2] L. Cheng and F. 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*, vol. 119, Article ID 102248, 2021.
- [3] L. Cheng and F. Meng, "Public key authenticated encryption with keyword search from LWE," in *Computer Security – ESORICS 2022*, pp. 303–324, Springer, Cham, 2022.
- [4] T. Chi, B. Qin, and D. Zheng, "An efficient searchable public-key authenticated encryption for cloud-assisted medical internet of things," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8816172, 11 pages, 2020.
- [5] K. Emura, "Generic construction of public-key authenticated encryption with keyword search revisited: Stronger security and efficient construction," in *APKC '22: Proceedings of the 9th ACM on ASIA Public-Key Cryptography Workshop*, pp. 39–49, Association for Computing Machinery, New York, NY, USA, May 2022.
- [6] Q. Huang and H. Li, "An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks," *Information Sciences*, vol. 403–404, pp. 1–14, 2017.
- [7] Z.-Y. Liu, Y.-F. Tseng, R. Tso, M. Mambo, and Y.-C. Chen, "Public-key authenticated encryption with keyword search: cryptanalysis, enhanced security, and quantum-resistant instantiation," in *ASIA CCS '22: Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, pp. 423–436, Association for Computing Machinery, New York, NY, USA, May 2022.
- [8] M. Noroozi and Z. Eslami, "Public key authenticated encryption with keyword search: revisited," *IET Information Security*, vol. 13, no. 4, pp. 336–342, 2019.
- [9] X. Pan and F. Li, "Public-key authenticated encryption with keyword search achieving both multi-ciphertext and multi-trapdoor indistinguishability," *Journal of Systems Architecture*, vol. 115, Article ID 102075, 2021.
- [10] B. Qin, Y. Chen, Q. Huang, X. Liu, and D. Zheng, "Public-key authenticated encryption with keyword search revisited: security model and constructions," *Information Sciences*, vol. 516, pp. 515–528, 2020.
- [11] B. Qin, H. Cui, X. Zheng, and D. Zheng, "Improved security model for public-key authenticated encryption with keyword search," in *Provable and Practical Security*, pp. 19–38, Springer, Cham, 2021.
- [12] L. Yao, J. Weng, A. Yang et al., "Scalable CCA-secure public-key authenticated encryption with keyword search from ideal lattices in cloud computing," *Information Sciences*, vol. 624, pp. 777–795, 2023.
- [13] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Advances in*

- Cryptology - EUROCRYPT 2004*, pp. 506–522, Springer, Berlin, Heidelberg, 2004.
- [14] J. Li, Y. Zhang, J. Ning, X. Huang, G. S. Poh, and D. Wang, “Attribute based encryption with privacy protection and accountability for cloudIoT,” *IEEE Transactions on Cloud Computing*, vol. 10, no. 2, pp. 762–773, 2022.
- [15] Y. Lu and J. Li, “Lightweight public key authenticated encryption with keyword search against adaptively-chosen-targets adversaries for mobile devices,” *IEEE Transactions on Mobile Computing*, vol. 21, no. 12, pp. 4397–4409, 2022.
- [16] Y. Lu, J. Li, and F. Wang, “Pairing-free certificate-based searchable encryption supporting privacy-preserving keyword search function for IIoTs,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2696–2706, 2021.
- [17] Y. Lu, J. Li, and Y. Zhang, “Privacy-preserving and pairing-free multirecipient certificateless encryption with keyword search for cloud-assisted IIoT,” *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2553–2562, 2020.
- [18] Y. Lu, J. Li, and Y. Zhang, “Secure channel free certificate-based searchable encryption withstanding outside and inside keyword guessing attacks,” *IEEE Transactions on Services Computing*, vol. 14, no. 6, pp. 2041–2054, 2021.
- [19] X. Liu, K. He, G. Yang, W. Susilo, J. Tonien, and Q. Huang, “Broadcast authenticated encryption with keyword search,” in *Information Security and Privacy*, pp. 193–213, Springer, Cham, 2021.
- [20] M. Ali, H. Ali, T. Zhong, F. Li, Z. Qin, and A. A. Ahmed Abdelrahman, “Broadcast searchable keyword encryption,” in *2014 IEEE 17th International Conference on Computational Science and Engineering*, pp. 1010–1016, IEEE, Chengdu, China, December 2014.
- [21] N. Attrapadung, J. Furukawa, and H. Imai, “Forward-secure and searchable broadcast encryption with short ciphertexts and private keys,” in *Advances in Cryptology – ASIACRYPT 2006*, pp. 161–177, Springer, Berlin, Heidelberg, 2006.
- [22] S. Chatterjee and S. Mukherjee, “Keyword search meets membership testing: adaptive security from SXDH,” in *Progress in Cryptology – INDOCRYPT 2018*, pp. 21–43, Springer, Cham, 2018.
- [23] T. Feng and J. Si, “Certificateless searchable encryption scheme in multi-user environment,” *Cryptography*, vol. 6, no. 4, Article ID 61, 2022.
- [24] P. Jiang, F. Guo, and Y. Mu, “Efficient identity-based broadcast encryption with keyword search against insider attacks for database systems,” *Theoretical Computer Science*, vol. 767, pp. 51–72, 2019.
- [25] A. Kiayias, O. Oksuz, A. Russell, Q. Tang, and B. Wang, “Efficient encrypted keyword search for multi-user data sharing,” in *Computer Security – ESORICS 2016*, pp. 173–195, Springer, Cham, 2016.
- [26] M. Ma, S. Fan, and D. Feng, “Multi-user certificateless public key encryption with conjunctive keyword search for cloud-based telemedicine,” *Journal of Information Security and Applications*, vol. 55, Article ID 102652, 2020.
- [27] N. Yang, Q. Zhou, Q. Huang, and C. Tang, “Multi-recipient encryption with keyword search without pairing for cloud storage,” *Journal of Cloud Computing*, vol. 11, Article ID 10, 2022.
- [28] K. Zhang, M. Wen, R. Lu, and K. Chen, “Multi-client sub-linear boolean keyword searching for encrypted cloud storage with owner-enforced authorization,” *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 6, pp. 2875–2887, 2021.
- [29] M. Ambrona, G. Barthe, and B. Schmidt, “Generic transformations of predicate encodings: constructions and applications,” in *Advances in Cryptology – CRYPTO 2017*, pp. 36–66, Springer, Cham, 2017.
- [30] J. Chen, R. Gay, and H. Wee, “Improved dual system ABE in prime-order groups via predicate encodings,” in *Advances in Cryptology–EUROCRYPT 2015*, pp. 595–624, Springer, Berlin, Heidelberg, 2015.
- [31] J. Chen and J. Gong, “ABE with tag made easy–concise framework and new instantiations in prime-order groups,” in *Advances in Cryptology–ASIACRYPT 2017*, pp. 35–65, Springer, Cham, 2017.
- [32] A. Barth, D. Boneh, and B. Waters, “Privacy in encrypted content distribution using private broadcast encryption,” in *Financial Cryptography and Data Security*, pp. 52–64, Springer, Berlin, Heidelberg, 2006.
- [33] N. Fazio and I. M. Perera, “Outsider-anonymous broadcast encryption with sublinear ciphertexts,” in *Public Key Cryptography – PKC 2012*, pp. 225–242, Springer, Berlin, Heidelberg, 2012.
- [34] A. Kiayias and K. Samari, “Lower bounds for private broadcast encryption,” in *Information Hiding*, pp. 176–190, Springer, Berlin, Heidelberg, 2012.
- [35] H. Kobayashi, Y. Watanabe, and J. Shikata, “Asymptotically tight lower bounds in anonymous broadcast encryption and authentication,” in *Cryptography and Coding*, pp. 105–128, Springer, Cham, 2021.
- [36] J. Li and J. Gong, “Improved anonymous broadcast encryptions: tight security and shorter ciphertext,” in *ACNS 2018 – 16th International Conference on Applied Cryptography and Network Security*, pp. 497–515, Springer-Verlag, Berlin, Heidelberg, July 2018.
- [37] B. Libert, K. G. Paterson, and E. A. Quaglia, “Anonymous broadcast encryption: adaptive security and efficient constructions in the standard model,” in *Public Key Cryptography – PKC 2012*, pp. 206–224, Springer, Berlin, Heidelberg, 2012.
- [38] M. Abdalla, M. Bellare, and G. Neven, “Robust encryption,” in *Theory of Cryptography*, pp. 480–497, Springer, Berlin, Heidelberg, 2010.
- [39] M. Abdalla, M. Bellare, and G. Neven, “Robust encryption,” *Journal of Cryptology*, vol. 31, no. 2, pp. 307–350, 2018.
- [40] D. Naor, M. Naor, and J. Lotspiech, “Revocation and tracing schemes for stateless receivers,” in *Advances in Cryptology – CRYPTO 2001*, pp. 41–62, Springer, Berlin, Heidelberg, 2001.
- [41] K. Emura, K. Kajita, and G. Ohtake, “Outsider-anonymous broadcast encryption with keyword search: generic construction, CCA security, and with sublinear ciphertexts,” *IACR Cryptology ePrint Archive*, Article ID 1119, 2023.
- [42] K. Emura, “Generic construction of broadcast authenticated encryption with keyword search,” *IACR Cryptology ePrint Archive*, Article ID 401, 2023.