Chosen Ciphertext Security via BARGs

Takahiro Matsuda

National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan t-matsuda@aist.go.jp

Abstract. In this paper, we show a new set of cryptographic primitives that generically leads to chosen ciphertext secure (CCA secure) public-key encryption (PKE). Specifically, we show how a (non-interactive, publicly verifiable) batch argument (BARG) for NP can be combined with a chosen plaintext secure PKE scheme to achieve a CCA secure one. The requirement of the succinctness of the proof size of a BARG in our result is rather mild: The proof size is $O(k^\epsilon)$ for some non-negative constant $\epsilon < 1$ when the correctness of k statements is simultaneously proved.

1 Introduction

The most basic security notion for public-key encryption (PKE) is indistinguishability against chosen plaintext attacks (CPA security, for short) [GM82]. Intuitively, CPA security guarantees that an adversary can obtain no information about a message from its encryption, except for its length. However, in practice, PKE schemes should satisfy the stronger notion of indistinguishability against chosen ciphertext attacks (CCA security, for short) [NY90,RS92]. CCA security implies non-malleability [DDN91,BDPR98], and provides security guarantees against active adversaries [Ble98].

Since CCA security is stronger than CPA security, the existence of CCA secure PKE implies that of CPA secure one. However, the implication of the opposite direction is not known. While a partial negative result was shown by Gertner, Malkin, and Myers [GMM07], the question whether a CCA secure PKE scheme can be constructed from a CPA secure one has still been standing as a major open question in cryptography.[1]

To gain insights towards solving the above open question, we tackle a related question: *When combined with a CPA secure PKE scheme, what additional generic primitive and/or security/structural property is sufficient to obtain a CCA secure PKE scheme?* A number of works have (implicitly or explicitly) tackled this question, and we review some of them in Section 1.2. The hope is that such an additional generic primitive and/or property that is sufficient for

[1] Interestingly, the results by Kitagawa and Matsuda [KM19,KM20] show that such a CPA-to-CCA transformation is possible for *key-dependent message (KDM) security* [BRS03] which considers security of encryptions that could be encrypting messages dependent on a secret key. In this paper, we focus on the standard *indistinguishability* notion. [GM82].

constructing a CCA secure PKE scheme in combination with a CPA secure one, and the techniques and ideas behind such constructions, will ultimately lead to answering the above fundamental open question.

In this paper, we will show that a *(non-interactive) batch argument system* for NP [KPY19,CJJ21], often abbreviated as "*BARG*", also serves as such an additional generic primitive for obtaining a CCA secure PKE scheme. A BARG is a non-interactive argument (i.e. computationally sound proof) system in which the correctness of multiple statements of an NP language can be proved in one shot, and furthermore it has a special succinctness property regarding the proof size: When the correctness of k statements each of which is of length at most n is simultaneously proved, the size of a proof π is *sublinear* in k, namely, $|\pi| = \mathsf{poly}(n, \lambda) \cdot o(k)$ where λ denotes the security parameter. Importantly, unlike a closely related primitive of non-interactive succinct argument system (SNARG) [GW11] whose proof size is sublinear in the length of the proved statement (and witness), the proof size of a BARG could be a polynomial of the length n of each statement whose correctness is being proved (and hence also polynomial in the length of the corresponding witness). Hence, in terms of the existence, BARGs are strictly weaker than SNARGs. In fact, while the security (soundness) of SNARGs cannot be proved from any assumption via black-box reductions [GW11], such an impossibility does not exist for BARGs, and a number of recent works have shown BARGs based on the hardness of various concrete number-theoretic problems (e.g. [CJJ21,KVZ21,CJJ22,HJKS22,WW22,KLVW23].).

We believe that it is natural to study the power of the combination of a CPA secure PKE scheme and a BARG for obtaining a CCA secure PKE scheme, since a BARG is one form of proof systems, and the well-known constructions of a CCA secure PKE scheme from the combination of a CPA secure PKE scheme and a non-interactive zero-knowledge (NIZK) proof system [NY90,DDN91] are also the combination of a CPA secure PKE scheme and a proof system. A natural question left from our work is what other generic primitive, and in particular whether a CPA secure PKE scheme, is sufficient for achieving a BARG.

1.1 Our Contribution

The main result of this paper can be informally stated as follows:

Theorem 1 ((Informal)). *If there exists a CPA secure PKE scheme and a BARG for NP with semi-adaptive soundness and k^ϵ-succinctness for some non-negative constant $\epsilon < 1$, then there exists a CCA secure PKE scheme.*

Here, the $\ell(k)$-succinctness means that the size of a proof π generated for proving the correctness of k statements each with length n by the BARG satisfies $|\pi| = \mathsf{poly}(n, \lambda) \cdot \ell(k)$, where λ denotes the security parameter[2], and semi-adaptive

[2] The definition of succinctness of most existing works on BARGs also require the succinctness of the size of a common reference string (CRS) and the running time of the verification algorithm. We do not pose these additional requirements (other than that the CRS size is $\mathsf{poly}(n, k.\lambda)$ and that the verification algorithm is PPT).

soundness is a soundness requirement specific to a BARG, which lies somewhere between non-adaptive and adaptive soundness. For the formal definitions for these properties, see Section 3.1.

We achieve our main result by constructing a *tag-based encryption (TBE)* scheme [Kil06] satisfying the security notion called indistinguishability against selective-tag weak chosen ciphertext attacks, which we simply abbreviate as *wCCA security*. Constructing a wCCA secure TBE scheme is sufficient for our purpose since it can be combined with a one-time signature scheme to obtain a CCA secure PKE.

More specifically, we construct a wCCA secure TBE scheme from the combination of a CPA secure PKE, a tag-based variant of an equivocal bit commitment scheme [DIO98], and a BARG, among which the second primitive can be generically built from any pseudorandom generator (PRG) and thus from any one-way function [HILL99], Technically, our proposed wCCA secure TBE construction is based on the constructions of CCA secure PKE by Koppula and Waters [KW19] and Kitagawa et al. [KMT19]. The former work constructs a CCA secure PKE scheme from the combination of a CPA secure PKE scheme and a special type of a PRG called a hinting PRG; The latter work is based on the Koppula-Waters construction, and can be seen to build a CCA secure PKE scheme from a CPA secure PKE scheme and a symmetric-key encryption (SKE) scheme satisfying KDM security [BRS03]. In both of the constructions [KW19,KMT19], the additional building blocks are used to check the validity of ciphertext components in some specific way. Our proposed construction can be seen to clarify that a BARG is sufficient for realizing the validity checking in a Koppula-Waters-type construction. We give a more detailed technical explanation of our proposed construction in Section 2. The formal description of our proposed wCCA secure TBE scheme as well as its security proof are given in Section 4.

On the Succinctness Requirement for a BARG. We note that the k^ϵ-succinctness property (with some constant $\epsilon < 1$) required in our result is arguably a very mild assumption, and satisfied by most of the existing BARG constructions [CJJ21,KVZ21,CJJ22,HJKS22,WW22,KLVW23]. In fact, some of the existing BARGs (e.g. [CJJ21,KLVW23]) achieve the proof size polylogarithmic in the batch size k. Furthermore, the recent work by Kalai et al. [KLVW23] showed how to transform a BARG with k^ϵ-succinctness (with any $\epsilon < 1$) into one whose proof size is polylogarithmic in k. However, the transformation of [KLVW23] requires some additional property for the building block BARG and furthermore uses an additional building block that is not known to be implied by a BARG alone (or from a CPA secure PKE scheme).

1.2 Related Work

CCA Secure PKE from Generic Cryptographic Primitives. The very first construction of a CCA secure PKE scheme was built from the combination of a CPA secure PKE scheme and a NIZK proof system by Dolev et al. [DDN91], based

on the earlier work by Naor and Yung [NY90] that achieved only non-adaptive CCA (CCA1) security.

CCA secure PKE can be built from any injective trapdoor function (TDF) as shown by Hohenberger et al. [HKW20], which subsumes the results of the earlier works that construct CCA secure PKE from TDF with additional properties such as lossy TDF [PW08], TDF secure under correlated products [RS09], and other types of TDF or related primitives [MY10,KMO10,Wee10,YYHK16]. Another line of works showed constructions of CCA secure PKE from the combination of a CPA secure PKE scheme with a deterministic function *without* a trapdoor but has some special security property: Specifically, as mentioned earlier, the construction by Koppula and Waters [KW19] combines a CPA secure PKE scheme with a hinting PRG; the construction by Matsuda and Hanaoka [MH14b] combines a CPA secure PKE scheme with a hash function called UCE (universal computational extractor) [BHK13] that tries to capture security properties a hash function satisfied by a random oracle in the standard model.

CCA secure PKE schemes can be built also from identity-based encryption [CHK04], and a weaker primitive of (wCCA secure) TBE [Kil06] (which we also use in this work). Another line of works [Dac14,MH16] showed how a PKE scheme with (standard-model, statistical) plaintext-awareness-1 [BP04,MSs12], which by itself only implies CCA1 security, can be used together with some other security properties to achieve a CCA secure PKE scheme. Several works [HK15,MH15,KMT19,KM19,KM20] showed the usefulness of KDM security [BRS03] for constructing a CCA secure PKE scheme. In particular, the most recent work in this direction by Kitagawa and Matsuda [KM20] showed that the combination of a CPA secure PKE scheme and a bit SKE scheme with circular security (which requires that each bit of a secret key can be securely encrypted) is sufficient for obtaining a CCA secure PKE scheme.

The works [SW14,MH14a] showed a connection of CCA secure PKE with cryptographic obfuscation [BGI$^+$01], namely, a construction of a CCA secure PKE scheme from indistinguishability obfuscation [SW14], and from an obfuscation of a multi-bit-output point function with a certain security property [MH14a].

Relation with Very Recent Works on "NIZK from BARGs". Very recently, Champion and Wu [CW23] and Bitansky et al. [BKP$^+$23] showed (among other things) how to construct a non-interactive zero-knowledge (NIZK) argument system for NP based on a BARG. Note that once a NIZK argument system is constructed from a BARG (possibly together with any primitive implied by a CPA secure PKE scheme), then we can use the NIZK argument system together with a CPA secure PKE scheme in the Naor-Yung paradigm [NY90,DDN91] to obtain a CCA secure PKE scheme, and hence our main result (Theorem 1) becomes just its corollary. However, we stress that our main result is not implied by these very recent works.

- The construction of a NIZK argument system by Champion and Wu [CW23] uses a dual-mode commitment scheme (which is known to be implied by

lossy encryption [BHY09]) and a local PRG (a PRG each of whose output bits depends on a small number of input bits) as additional building blocks. However, these additional primitives are not known to be constructed from the combination of a CPA secure PKE scheme and a BARG. Furthermore, the succinctness requirement on the underlying BARG is stronger than ours (the proof size has to be polylogarithmic in the batch size k).

– There are mainly two constructions of a NIZK argument system in the work by Bitansky et al. [BKP+23].
  - The first construction uses only a BARG and a one-way function as building blocks, and furthermore the succinctness requirement on the underlying BARG is the same as ours (k^ϵ-succinctness with any constant $\epsilon < 1$). However, although the zero-knowledge property achieved by their construction is statistical (i.e. its guarantee is against computationally unbounded adversaries), it is only inverse-polynomial, meaning that the upper bound of the advantage of an adversary in breaking the zero-knowledge property of their construction is bounded only by $1/p$ for a pre-determined polynomial $p = p(\lambda)$ that is hard-coded in their construction. (We can choose any polynomial p, only after which their NIZK argument construction is defined.) Furthermore, in this construction, the prover algorithm is a non-uniform algorithm (i.e. it needs to use some parameter determined by the security parameter λ as an additional input). Actually, the prover algorithm of this construction can be uniform in the so-called distributional setting, namely, for the NP relation supported by their construction, there exists an efficietly samplable distribution such that the ZK property holds only when a statement/witness pair comes from the distribution. However, this result is existential and not constructive, namely, their result for the uniform prover does not tell us a concrete distribution with which the ZK property is satisfied, due to their proof technique.
  - Their second construction is a NIZK argument system with ordinary zero-knowledge property (i.e. an adversary's advantage can be bounded to be negligible), however, this construction uses a dual-mode commitment scheme similarly to [CW23]. We note that it is not known whether a dual-mode commitment scheme necessary for their result is implied by a CPA secure PKE scheme.[3]

1.3 Paper Organization

In Section 2, we give a technical overview of our result. In Section 3, we review the basic notation and the definitions for primitives treated in this paper. In Section 4, we show our main technical result: a wCCA secure TBE scheme based on the combination of a CPA secure PKE scheme and a BARG.

[3] Dual-mode commitment schemes can be constructed from lossy encryption, but the latter cannot be constructed from a CPA secure PKE scheme due to the black-box separation between a PKE scheme and an oblivious transfer protocol (OT), and lossy encryption implies OT.

2 Technical Overview

In this section, we give a technical overview of our proposed construction of a wCCA secure TBE scheme based on a CPA secure PKE scheme and a BARG.

We first quickly recall that TBE is an extension of PKE where the encryption and decryption algorithms take a bit-string called a tag as an additional input [Kil06]. Throughout the paper, a tag will be denoted by $\mathtt{t}$. CCA security for TBE is also defined by extending CCA security for PKE so that an adversary sends a tag in addition to a ciphertext when making a decryption query. *Weak* CCA (wCCA) security considers adversaries that do not make a decryption query containing the challenge tag $\mathtt{t}^*$ under which the challenge ciphertext is generated. For the formal definitions for TBE, see Section 3.3.

As mentioned in Section 1.1, our proposed wCCA secure TBE construction is based on the constructions of [KW19,KMT19], and is built from the combination of a CPA secure PKE scheme, a tag-based equivocal bit commitment scheme (TBEQC, for short), and a k^ϵ-succinct BARG for NP (with constant $\epsilon < 1$), among which the second building block can be built from a PRG (and from any CPA secure PKE scheme). In Section 4, we give a direct construction of a wCCA secure TBE scheme from these building blocks. However, to explain the ideas on how our wCCA secure TBE construction works, it is useful to introduce a special type of 1-bit TBE scheme as an intermediate building block, which itself is not wCCA secure but has several useful properties, and then describe our proposed wCCA secure TBE using this 1-bit TBE scheme, and thus we take such an approach in this section.

In the rest of this section, we explain the following.

- A TBEQC scheme and its required properties.
- The "base" 1-bit TBE scheme based on a CPA secure PKE scheme and a TBEQC scheme.
- The proposed wCCA secure TBE scheme based on the base 1-bit TBE scheme and a k^ϵ-succinct BARG.

Tag-Based Equivocal Bit Commitment. Informally, a TBEQC scheme is a tag-based bit commitment scheme with the two modes of operation: the normal mode and the EQ (equivocation) mode.

- When the EQ mode is setup, one specifies a special equivocal tag $\mathtt{t}^*$, and then an equivocal commitment key $\widetilde{ck}$ is generated together with an equivocal commitment $\widetilde{\gamma}$ and a pair of randomnesses $\widetilde{\rho}^0$ and $\widetilde{\rho}^1$ such that $\widetilde{\rho}^s$ is consistent with $\widetilde{\gamma}$ being consistent as a commitment of the message bit s under the key $\widetilde{ck}$ and the tag $\mathtt{t}^*$, for both $s \in \{0,1\}$. The normal mode and the EQ mode must be computationally indistinguishable, even taking the randomness used for generating a commitment into account. That is, for any equivocal tag $\mathtt{t}^*$ and a message bit $s \in \{0,1\}$, the tuple (ck, γ, ρ) where ck is a normal commitment key, γ is a commitment of s under $\mathtt{t}^*$ and ck, and ρ is the randomness used for generating γ, is indistinguishable from the tuple $(\widetilde{ck}, \widetilde{\gamma}, \widetilde{\rho}^s)$ generated in the EQ mode set up using $\mathtt{t}^*$.

– On the other hand, statistical binding must still hold even in the EQ mode, under all tags t with $\mathsf{t} \neq \mathsf{t}^*$. That is, except with a negligible probability over the set up of the EQ mode (with t^*) that generates an equivocal commitment key $\widetilde{ck}$, no commitment γ generated under $\widetilde{ck}$ and under any tag $\mathsf{t} \neq \mathsf{t}^*$ can be opened to the message bit $s = 0$ and message bit $s = 1$ simultaneously.

For the formal definition, see Section 3.4, where we also explain how it can be constructed from any PRG.

"Base 1-bit TBE" from CPA Secure PKE and TBEQC. We now explain how the base 1-bit TBE scheme, which we will use as an intermediate building block for explaining our wCCA secure TBE scheme below, is constructed from the combination of a CPA secure PKE scheme and a TBEQC scheme.

– A public key of the base 1-bit TBE scheme is of the form (pk^0, pk^1, ck), where each pk^v $(v \in \{0,1\})$ is a public key of the CPA secure PKE scheme, and ck is a commitment key of the TBEQC scheme. The secret key of the base 1-bit TBE scheme is sk^0 corresponding to pk^0.
– To encrypt a plaintext bit $s \in \{0,1\}$ under a tag t, we proceed as follows:
  1. Pick two randomnesses r^0, r^1 of the CPA secure PKE scheme, and two randomnesses ρ^0, ρ^1 of the TBEQC scheme.
  2. Compute two PKE-ciphertexts c^0 and c^1 as follows

$$\left(c^0,\ c^1 \right) \leftarrow \begin{cases} \left(\mathsf{Enc}(pk^0, \rho^0; r^0),\ \mathsf{Enc}(pk^1, \perp; r^1) \right) & \text{if } s = 0 \\ \left(\mathsf{Enc}(pk^0, \perp; r^0),\ \mathsf{Enc}(pk^1, \rho^1; r^1) \right) & \text{if } s = 1 \end{cases},$$

     where $\mathsf{Enc}(pk, m; r)$ denotes an encryption of a plaintext m using a randomness r under a public key pk of the CPA secure PKE scheme, and $\perp$ denotes an invalidity symbol (unambiguously encoded in the plaintext space).
  3. Compute a commitment γ of s under ck and t using ρ^s as the randomness by the TBEQC scheme. We denote this process by $\gamma = \mathsf{Com}(ck, \mathsf{t}, s; \rho^s)$.
  4. The resulting ciphertext C of the base 1-bit TBE scheme is of the form $C = (c^0, c^1, \gamma)$.

To decrypt a ciphertext $C = (c^0, c^1, \gamma)$ under a tag t and recover the encrypted bit s, we compute

$$s = \begin{cases} 0 & \text{if } \mathsf{Dec}(sk^0, c^0) = \rho^0 \neq \perp \wedge \mathsf{Com}(ck, \mathsf{t}, 0; \rho^0) = \gamma \\ 1 & \text{otherwise} \end{cases}, \tag{1}$$

where $\mathsf{Dec}(sk, c)$ denotes the decryption of a ciphertext c using a secret key sk.

The confidentiality of the plaintext s encrypted in C in case there is no decryption oracle, can be argued as follows. Suppose an adversary specifies the challenge tag t^* under which the challenge ciphertext is generated. By the indistinguishability of the normal mode and EQ mode of the TBEQC scheme, we can

indistinguishably switch the generation of ck, γ, and ρ^0, ρ^1 into $(\widetilde{ck}, \widetilde{\gamma}, \widetilde{\rho}^0, \widetilde{\rho}^1)$ generated in the EQ mode under t^*. Then, we generate both of the PKE-ciphertexts c^v $(v \in \{0,1\})$ by $\mathsf{Enc}(pk^v, \widetilde{\rho}^v; r^v)$, which an adversary cannot notice due to the CPA security under the public key pk^{1-s}. The ciphertext $C = (c^0, c^1, \widetilde{\gamma})$ generated in this way is now independent of a plaintext bit s.

An important property of the above base 1-bit TBE construction is that it has the special procedure for checking the "well-formedness" of a ciphertext C, using a "witness" w that can be generated from the plaintext s and the randomness $(r^0, r^1, \rho^0, \rho^1)$ used to generate C. Specifically, we define the witness w by $w = (s, r^s, \rho^s)$. Then, to check the well-formedness of a ciphertext $C = (c^0, c^1, \gamma)$ under a tag t using the witness $w = (s, r^s, \rho^s)$, we check whether the following is satisfied simultaneously:

$$\mathsf{Enc}(pk^s, \rho^s; r^s) = c^s \qquad \text{and} \qquad \mathsf{Com}(ck, \mathsf{t}, s; \rho^s) = \gamma. \tag{2}$$

Importantly, if C satisfies Eq. (2), then C can be decrypted by the following alternative procedure using the other secret key sk^1 (corresponding to pk^1):

$$s = \begin{cases} 1 & \text{if } \mathsf{Dec}(sk^1, c^1) = \rho^1 \neq \bot \wedge \mathsf{Com}(ck, \mathsf{t}, 1; \rho^1) = \gamma \\ 0 & \text{otherwise} \end{cases}. \tag{3}$$

The decryption result with Eq. (1) and that with Eq. (3) agree except with negligible probability, due to the statistical binding of the TBEQC scheme. In fact, due to the statistical binding in the EQ mode, this "interchangeability" of sk^0 and sk^1 continues to work even if we setup the commitment key ck by the EQ mode, as long as the decryption is performed under tags t that are different from the equivocal tag t^* used for setting up of the EQ mode.

Finally, the key feature of the base 1-bit TBE construction is that the above argument on confidentiality continues to work even against an adversary that is

- given the witness $w = (s, r^s, \rho^s)$ for the challenge ciphertext, and
- even in the situation where the adversary has access to the "constrained" decryption oracle that takes a pair $(\mathsf{t}, C = (c^0, c^1, \gamma))$ with $\mathsf{t} \neq \mathsf{t}^*$ as input, and decrypt C under t only when C is guaranteed to be "well-formed", i.e. there exists a witness $w = (s, r^s, \rho^s)$ satisfying Eq. (2).

This is due to the above explained property of the "interchangeability" of sk^0 and sk^1, and the fact that the witness w does not contain r^{1-s}, in which case we can rely on the CPA security of the PKE scheme under pk^{1-s}. Hence, if we can somehow guarantee that an adversary can observe only the decryption result of well-formed ciphertexts, we can show the wCCA security of this base 1-bit TBE construction, and we indeed guarantee this by using a BARG, as explained next.

wCCA Secure TBE from "Base 1-bit TBE" and BARG. Here, we explain how the above base 1-bit TBE scheme can be combined with a k^ϵ-succinct BARG (with $\epsilon < 1$) to finally construct a wCCA secure TBE scheme.

The public key of our proposed TBE construction is a public key of the base 1-bit TBE scheme[4], a common reference string (CRS) of the BARG, and a universal hash function H, and the secret key is that of the base 1-bit TBE scheme.

The encryption procedure for encrypting a plaintext m under a tag $\mathtt{t}$ in our proposed TBE construction proceeds as follows:

1. First, pick a random string $\mathbf{s} = (s_1, \ldots, s_\kappa) \in \{0,1\}^\kappa$, where $\kappa = \kappa(\lambda)$ is some polynomial of λ to be explained later, and λ denotes the security parameter.
2. For each $i \in [\kappa]$, encrypt the i-th bit $s_i \in \{0,1\}$ under the tag $\mathtt{t}$ by the base 1-bit TBE scheme. Let C_i denote the resulting bit-encryption of s_i. In this step, from the randomness used to generate C_i, we also generate the witness w_i for the fact that "C_i is well-formed" for each $i \in [\kappa]$. As seen above, the base 1-bit TBE construction has such a witness.
3. Generate a BARG proof π proving that every C_i is well-formed, using the witnesses $(w_i)_{i\in[\kappa]}$. We can ensure that the length of each statement and witness is $\mathsf{poly}(\lambda)$, independent of the batch size κ. Hence, by the k^ϵ-succinctness of the BARG, there exists some polynomial $p = p(\lambda)$ such that we have $|\pi| \leq p \cdot \kappa^\epsilon$.
4. Mask the plaintext m by $c_m = H(\mathbf{s}) \oplus m$.
5. The resulting ciphertext $C_{\mathtt{wCCA}}$ of the proposed wCCA secure TBE scheme is of the form $C_{\mathtt{wCCA}} = ((C_i)_{i\in[\kappa]}, \pi, c_m)$.

To decrypt $C_{\mathtt{wCCA}} = ((C_i)_{i\in[\kappa]}, \pi, c_m)$ under a tag $\mathtt{t}$, we proceed as follows:

1. Firstly, check the validity of the BARG proof π. If π is rejected, then output the invalidity symbol $\perp$ and terminate. Otherwise (i.e. π is accepted), proceed to the next step. Intuitively, when π is accepted, the (semi-adaptive) soundness of the BARG guarantees that every C_i is well-formed.
2. For each $i \in [\kappa]$, decrypt each C_i under $\mathtt{t}$ by the decryption procedure of the base 1-bit TBE scheme, and recover a bit s_i.
3. Set $\mathbf{s} = (s_1, \ldots, s_\kappa)$.
4. Recover the plaintext m by $m = c_m \oplus H(\mathbf{s})$, and output m.

A high-level idea of why this construction can be proved wCCA secure is as follows: Note that due to the verification of the BARG proof π in the decryption procedure, a wCCA adversary against the above scheme can observe the decryption results of $(C_i)_{i\in[\kappa]}$ only in case these components are well-formed. Hence, by the confidentiality property of the base 1-bit TBE scheme explained above, we can switch (together with the commitment key contained in the public key) to the situation where each ciphertext C_i is independent of the information of

[4] Strictly speaking, in order for the following argument to go through, we have to generate a public key of the base 1-bit TBE scheme for each position $i \in [\kappa]$. We allow ourselves to be inaccurate about this point in this technical overview. In the actual construction in Section 4, we in fact adopt a slight optimization to reuse the same public key pair (pk^0, pk^1) for each position, so that only the commitment key is generated for each position.

the bit s_i, even if the witness w_i for checking the well-formedness of C_i is visible to the adversary, and even if the adversary has access to the decryption oracle (of the wCCA security experiment of the proposed TBE construction). Then, the components $(C_i)_{i\in[\kappa]}$ essentially do not leak the information of $\mathbf{s} \in \{0,1\}^\kappa$, and the only components in $C_{\mathtt{wCCA}}$ that are still dependent on $\mathbf{s}$ are the BARG proof π and the message-masking component c_m. Here, when c_m is excluded, the (min-)entropy of $\mathbf{s}$ given π is essentially $|\mathbf{s}| - |\pi| \geq \kappa - p \cdot \kappa^\epsilon$. Hence, by choosing κ to be sufficiently large (depending on ϵ and p), we can guarantee that $\mathbf{s}$ has sufficiently large entropy even given π, so that the leftover hash lemma [HILL99,DRS04] implies that $c_m = H(\mathbf{s}) \oplus m$ statistically hides m.

As mentioned earlier, in Section 4, the proposed wCCA secure TBE construction is described in such a way that it is directly built from a CPA secure PKE scheme, a TBEQC scheme, and a BARG. However, ideas behind the proposed construction and our security proof rely on the explanations in this section. For all the details, see Section 4.

3 Preliminaries

In this section, we review the basic notation and the definitions for primitives.

Basic Notation. $\mathbb{N}$ denotes the set of all natural numbers. For $n \in \mathbb{N}$, we define $[n] := \{1, \ldots, n\}$. If x and y are strings, "$|x|$" denotes the bit-length of x, and "$(x \stackrel{?}{=} y)$" is defined to be 1 if $x = y$ and 0 otherwise. If S is a finite set, then "$x \leftarrow S$" denotes that x is chosen uniform from S. If M is a probabilistic algorithm, then "$y \leftarrow \mathsf{M}(x; r)$" denotes that M computes y as output by taking x as input and using r as randomness. If we need not make the randomness used in M explicit, then we write it as "$y \leftarrow \mathsf{M}(x)$". If furthermore $\mathcal{O}$ is a function or an algorithm, then "$\mathsf{M}^{\mathcal{O}}$" denotes that M has oracle access to $\mathcal{O}$. A function $f(\cdot) : \mathbb{N} \to [0,1]$ is *negligible* if for all positive polynomials $p(\lambda)$ and all sufficiently large $\lambda \in \mathbb{N}$, we have $f(\lambda) < 1/p(\lambda)$. "$\mathsf{poly}$" denotes an unspecified positive polynomial. "PPT" stands for *probabilistic polynomial time*.

3.1 Non-interactive Batch Arguments

Here, we recall the definitions for a BARG. The definitions below are based on [KLVW23], which we slightly customize (simplify) so that it is sufficient for our purpose.

Let $\mathcal{L} \subseteq \{0,1\}^*$ be an NP language. A non-interactive batch argument system in the common reference string (CRS) model (abbreviated simply as *BARG*) for $\mathcal{L}$ consists of the three PPT algorithms $(\mathsf{CRSG}, \mathsf{BProve}, \mathsf{BVerify})$ with the following syntax:

CRS Generation:	$crs \leftarrow \mathsf{CRSG}(1^\lambda, k, n)$
Proof Generation:	$\pi \leftarrow \mathsf{BProve}(crs, (x_i)_{i\in[k]}, (w_i)_{i\in[k]})$
Verification:	$\top \,/\, \bot \leftarrow \mathsf{BVerify}(crs, (x_i)_{i\in[k]}, \pi)$

where $\mathsf{BVerify}$ is deterministic, crs is a CRS, $k : \mathbb{N} \to \mathbb{N}$ is a polynomial specifying the batch size, $n : \mathbb{N} \to \mathbb{N}$ is the upper bound on the statement length, each x_i in BProve is a statement that belongs to $\mathcal{L} \cap \{0,1\}^{n(\lambda)}$, each w_i is a witness for $x_i \in \mathcal{L}$, and π is a proof. Each x_i in $\mathsf{BVerify}$ could belong to $\{0,1\}^{n(\lambda)} \setminus \mathcal{L}$.

For correctness (completeness) of a BARG, we require that for all $\lambda \in \mathbb{N}$, all polynomials $k = k(\lambda)$ and $n = n(\lambda)$, all k valid statement/witness pairs $(x_1, w_1), \ldots, (x_k, w_k) \in (\mathcal{L} \cap \{0,1\}^n) \times \{0,1\}^*$, if $crs \leftarrow \mathsf{CRSG}(1^\lambda, k, n)$ and $\pi \leftarrow \mathsf{BProve}(crs, (x_i)_{i\in[k]}, (w_i)_{i\in[k]})$, then $\mathsf{BVerify}(crs, (x_i)_{i\in[k]}, \pi) = \top$.

Semi-adaptive Soundness. Let $\mathcal{P} = (\mathsf{CRSG}, \mathsf{BProve}, \mathsf{BVerify})$ be a BARG for an NP language $\mathcal{L}$. For $\mathcal{P}$ and an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, we define the semi-adaptive soundness experiment $\mathsf{Expt}^{\mathtt{sa\text{-}sound}}_{\mathcal{P},\mathcal{A}}(\lambda)$ as follows.

$\mathsf{Expt}^{\mathtt{sa\text{-}sound}}_{\mathcal{P},\mathcal{A}}(\lambda)$:
$(k, n, i^*, \mathsf{st}) \leftarrow \mathcal{A}_1(1^\lambda)$ // k and n are polynomials, $i^* \in [k(\lambda)]$
$crs \leftarrow \mathsf{CRSG}(1^\lambda, k, n)$
$((x_i)_{i\in[k(\lambda)]}, \pi) \leftarrow \mathcal{A}_2(crs, \mathsf{st})$
If **(a)** $\wedge$ **(b)** $\wedge$ **(c)** then return 1 else return 0:
(a) $\mathsf{BVerify}(crs, (x_i)_{i\in[k(\lambda)]}, \pi) = \top$
(b) $x_{i^*} \notin \mathcal{L}$
(c) $\forall i \in [k(\lambda)] : |x_i| \leq n(\lambda)$

Definition 1. *We say that a BARG $\mathcal{P}$ satisfies semi-adaptive soundness if for all PPT adversaries $\mathcal{A}$, $\mathsf{Adv}^{\mathtt{sa\text{-}sound}}_{\mathcal{P},\mathcal{A}}(\lambda) := \Pr[\mathsf{Expt}^{\mathtt{sa\text{-}sound}}_{\mathcal{P},\mathcal{A}}(\lambda) = 1]$ is negligible.*

Succinctness of the Proof Size. One of the key features required for a BARG is *succinctness*. Informally, we require the proof size of a BARG to be sublinear in the batch size k.[5]

Definition 2. *Let $\ell : \mathbb{N} \to \mathbb{N}$. We say that a BARG $(\mathsf{CRSG}, \mathsf{BProve}, \mathsf{BVerify})$ is* ℓ-succinct *if the following holds: If crs is generated as $crs \leftarrow \mathsf{CRSG}(1^\lambda, k, n)$, then a proof π generated by BProve satisfies $|\pi| = \mathsf{poly}(\lambda, n) \cdot \ell(k)$.*

In this paper, we will require arguably a very mild succinctness requirement for a BARG: We require it to be k^ϵ-succinct for some non-negative constant $\epsilon < 1$. We stress that our proposed construction works with any constant $\epsilon < 1$.

3.2 Public-Key Encryption

A public-key encryption (PKE) scheme consists of the three PPT algorithms $(\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ with the following syntax:

Key Generation:	$(pk, sk) \leftarrow \mathsf{KG}(1^\lambda)$
Encryption:	$c \leftarrow \mathsf{Enc}(pk, m)$
Decryption:	$m \,/\, \perp \leftarrow \mathsf{Dec}(sk, c)$

[5] In [KLVW23], "ℓ-succinctness" puts some requirements on the size of crs and the running time of $\mathsf{BVerify}$. We do not introduce them since they are unnecessary for our result.

where Dec is deterministic, (pk, sk) is a public/secret key pair, and c is a ciphertext of a plaintext m under pk.

We require that for all $\lambda \in \mathbb{N}$, all (pk, sk) output by $\mathsf{KG}(1^\lambda)$, and all m, we have $\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m$.

CCA and CPA Security. For a $\Pi = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ and an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, we define the CCA experiment $\mathsf{Expt}^{\mathtt{CCA}}_{\Pi,\mathcal{A}}(\lambda)$ as follows:

$\mathsf{Expt}^{\mathtt{CCA}}_{\Pi,\mathcal{A}}(\lambda)$:
$(pk, sk) \leftarrow \mathsf{KG}(1^\lambda)$
$(m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}_1^{\mathsf{Dec}(sk,\cdot)}(pk)$
$b \leftarrow \{0, 1\}$
$c^* \leftarrow \mathsf{Enc}(pk, m_b)$
$b' \leftarrow \mathcal{A}_2^{\mathsf{Dec}(sk,\cdot)}(c^*, \mathsf{st})$
Return $(b' \stackrel{?}{=} b)$.

In the experiment, it is required that $|m_0| = |m_1|$, and $\mathcal{A}_2$ is not allowed to submit the challenge ciphertext c^* to the decryption oracle. We also define the CPA experiment $\mathsf{Expt}^{\mathtt{CPA}}_{\Pi,\mathcal{A}}(\lambda)$ in the same way as the CCA experiment, except that $\mathcal{A}$ is not given access to the decryption oracle.

Definition 3. *We say that a PKE scheme Π is CCA secure if for all PPT adversaries $\mathcal{A}$, $\mathsf{Adv}^{\mathtt{CCA}}_{\Pi,\mathcal{A}}(\lambda) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{CCA}}_{\Pi,\mathcal{A}}(\lambda) = 1] - 1/2|$ is negligible. CPA security of a PKE scheme is defined analogously, using the CPA experiment.*

3.3 Tag-Based Encryption

Tag-based encryption (TBE) [Kil06] is a simple extension of PKE, where the encryption/decryption algorithms take an additional input called a "tag", and the correctness of decryption is guaranteed in case the encryption/decryption algorithms use the same tag.

A TBE scheme consists of the three algorithms $(\mathsf{TKG}, \mathsf{TEnc}, \mathsf{TDec})$ with the following syntax:

Key Generation:	$(pk, sk) \leftarrow \mathsf{TKG}(1^\lambda)$
Encryption:	$c \leftarrow \mathsf{TEnc}(pk, \mathsf{t}, m)$
Decryption:	$m \,/\, \perp \leftarrow \mathsf{TDec}(sk, \mathsf{t}, c)$

where TDec is deterministic, (pk, sk) is a public/secret key pair, $\mathsf{t} \in \{0,1\}^\lambda$ is a tag, and c is a ciphertext encrypting a plaintext m under pk and t.

We require that for all $\lambda \in \mathbb{N}$, all (pk, sk) output by $\mathsf{TKG}(1^\lambda)$, all $\mathsf{t} \in \{0,1\}^\lambda$, and all m, if $c \leftarrow \mathsf{TEnc}(pk, \mathsf{t}, m)$, then we have $\mathsf{TDec}(sk, \mathsf{t}, c) = m$.

Weak CCA Security. Here, we recall the security definition for TBE called *indistinguishability against selective-tag weak chosen ciphertext attacks* [Kil06], which we call *wCCA security*, for short.

For a TBE scheme $\mathcal{T} = (\mathsf{TKG}, \mathsf{TEnc}, \mathsf{TDec})$ and an adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$, we define the wCCA security experiment $\mathsf{Expt}^{\mathtt{wCCA}}_{\mathcal{T},\mathcal{A}}(\lambda)$ as follows:

$\mathsf{Expt}^{\mathtt{wCCA}}_{\mathcal{T},\mathcal{A}}(\lambda)$:
$(\mathsf{t}^*, \mathsf{st}_0) \leftarrow \mathcal{A}_0(1^\lambda)$
$(pk, sk) \leftarrow \mathsf{TKG}(1^\lambda)$
$(m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}_1^{\mathsf{TDec}(sk\cdot,\cdot)}(pk, \mathsf{st}_0)$
$b \leftarrow \{0,1\}$
$c^* \leftarrow \mathsf{TEnc}(pk, \mathsf{t}^*, m_b)$
$b' \leftarrow \mathcal{A}_2^{\mathsf{TDec}(sk,\cdot,\cdot)}(c^*, \mathsf{st})$
Return $(b' \stackrel{?}{=} b)$.

In the experiment, it is required that $|m_0| = |m_1|$, and $\mathcal{A}_1$ and $\mathcal{A}_2$ are not allowed to submit a decryption query (t, c) satisfying $\mathsf{t} = \mathsf{t}^*$.

Definition 4. *We say that a TBE scheme $\mathcal{T}$ is wCCA secure if for all PPT adversaries $\mathcal{A}$, $\mathsf{Adv}^{\mathtt{wCCA}}_{\mathcal{T},\mathcal{A}}(\lambda) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{wCCA}}_{\mathcal{T},\mathcal{A}}(\lambda) = 1] - 1/2|$ is negligible.*

Connection to CCA Secure PKE. A wCCA secure TBE scheme can be combined with a one-time signature scheme to construct a CCA secure PKE scheme [Kil06]. Hence, to obtain a CCA secure PKE scheme, it is sufficient to construct a wCCA secure TBE scheme.

Theorem 2 ([Kil06]). *If there exists a wCCA secure TBE scheme, then there exists a CCA secure PKE scheme.*

3.4 Tag-Based Equivocal Bit Commitment

Here, we introduce a definition of a tag-based variant of an equivocal bit commitment scheme [DIO98], which we abbreviate as TBEQC. As explained in Section 2, this primitive is useful for describing the Koppula-Waters type construction of PKE, and hence our proposed construction of a TBE scheme in Section 4.

A TBEQC scheme consists of the three PPT algorithms $(\mathsf{CKG}, \mathsf{Com}, \mathsf{EQSetup})$ with the following syntax:

Key Generation:	ck	$\leftarrow \mathsf{CKG}(1^\lambda)$
Commitment Generation:	γ	$\leftarrow \mathsf{Com}(ck, \mathsf{t}, s)$
EQ Mode Setup:	$(\widetilde{ck}, \widetilde{\gamma}, \widetilde{\rho}_0, \widetilde{\rho}_1)$	$\leftarrow \mathsf{EQSetup}(1^\lambda, \mathsf{t}^*)$

where ck is a commitment key, γ is a commitment of a message bit $s \in \{0,1\}$ under ck and a tag $\mathsf{t} \in \{0,1\}^\lambda$; $\widetilde{ck}$ and $\widetilde{\gamma}$ are an equivocal commitment key and an equivocal commitment, respectively, generated for a special (equivocal) tag $\mathsf{t}^* \in \{0,1\}^\lambda$; $\widetilde{\rho}_s$ is a randomness (for Com) that is consistent with $\widetilde{\gamma}$ in the sense that $\widetilde{\gamma}$ looks like a commitment of $s \in \{0,1\}$ geuerated by using $\widetilde{\rho}_s$ as a randomness, under $\widetilde{ck}$ and t^*.

For a TBEQC scheme, we require that for all $\lambda \in \mathbb{N}$, $s \in \{0,1\}$, and $\mathsf{t}^* \in \{0,1\}^\lambda$, if $(\widetilde{ck}, \widetilde{\gamma}, \widetilde{\rho}_0, \widetilde{\rho}_1) \leftarrow \mathsf{EQSetup}(1^\lambda, \mathsf{t}^*)$, then we have $\mathsf{Com}(\widetilde{ck}, \mathsf{t}^*, s; \widetilde{\rho}_s) = \widetilde{\gamma}$.

Security. We require two kinds of security properties for a TBEQC scheme: *indistinguishability of the two modes* and *statistical binding in the EQ mode.*

Definition 5. *Let* $\mathcal{C} = (\mathsf{CKG}, \mathsf{Com}, \mathsf{EQSetup})$ *be a TBEQC scheme in which the randomness space of* Com *is* $\mathcal{R}$*. We say that* $\mathcal{C}$ *is secure if it satisfies the following two security properties:*

- ***(Indistinguishability of the two modes)*** *For all PPT adversaries* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*,*

$$\mathsf{Adv}^{\mathtt{ind}}_{\mathcal{C},\mathcal{A}}(\lambda) := \left|\Pr[\mathsf{Expt}^{\mathtt{normal}}_{\mathcal{C},\mathcal{A}}(\lambda) = 1] - \Pr[\mathsf{Expt}^{\mathtt{eq}}_{\mathcal{C},\mathcal{A}}(\lambda) = 1]\right|$$

  is negligible, where the experiments $\mathsf{Expt}^{\mathtt{normal}}_{\mathcal{C},\mathcal{A}}(\lambda)$ *and* $\mathsf{Expt}^{\mathtt{eq}}_{\mathcal{C},\mathcal{A}}(\lambda)$ *are defined as follows:*

$\mathsf{Expt}^{\mathtt{normal}}_{\mathcal{C},\mathcal{A}}(\lambda)$:	$\mathsf{Expt}^{\mathtt{eq}}_{\mathcal{C},\mathcal{A}}(\lambda)$:
$(\mathsf{t}^*, s, \mathsf{st}) \leftarrow \mathcal{A}_1(1^\lambda)$	$(\mathsf{t}^*, s, \mathsf{st}) \leftarrow \mathcal{A}_1(1^\lambda)$
$ck \leftarrow \mathsf{CKG}(1^\lambda)$	$(\widetilde{ck}, \widetilde{\gamma}, \widetilde{\rho}_0, \widetilde{\rho}_1) \leftarrow \mathsf{EQSetup}(1^\lambda, \mathsf{t}^*)$
$\rho \leftarrow \mathcal{R}$	
$\gamma \leftarrow \mathsf{Com}(ck, \mathsf{t}^*, s; \rho)$	
$b' \leftarrow \mathcal{A}_2(ck, \gamma, \rho, \mathsf{st})$	$b' \leftarrow \mathcal{A}_2(\widetilde{ck}, \widetilde{\gamma}, \widetilde{\rho}_s, \mathsf{st})$
Return b'.	*Return* b'.

- ***(Statistical binding in the EQ mode)*** *The following quantity is negligible in* λ*:*

$$\max_{\mathsf{t}^* \in \{0,1\}^\lambda} \Pr_{(\widetilde{ck},\widetilde{\gamma},\widetilde{\rho}_0,\widetilde{\rho}_1) \leftarrow \mathsf{EQSetup}(1^\lambda, \mathsf{t}^*)} \left[\begin{array}{l} \exists \rho_0, \rho_1 \in \mathcal{R} \text{ and } \mathsf{t} \neq \mathsf{t}^* \text{ s.t.} \\ \mathsf{Com}(\widetilde{ck}, \mathsf{t}, 0; \rho_0) = \mathsf{Com}(\widetilde{ck}, \mathsf{t}, 1; \rho_1) \end{array}\right].$$

Instantiation from PRG. It is possible to construct a TBEQC scheme from any PRG (and hence from any one-way function [HILL99]). Specifically, such a construction appears as an "internal structure" of the constructions of a CCA secure PKE scheme in [KW19,KMT19], which can be seen as a natural extension of a (non-tag-based) equivocal commitment scheme based on Naor's commitment scheme [DIO98,Nao91] such that it supports tags. For completeness, we provide its description and security proof in Appendix A.

Theorem 3. *If there exists a one-way function, then there exists a secure TBEQC scheme satisfying the two security properties (i.e. indistinguishability of the two modes and statistical binding in the EQ mode).*

3.5 Universal Hash Family and Leftover Hash Lemma

Average Min-entropy. Let $(\mathcal{X}, \mathcal{Y})$ be a joint distribution. The *average min-entropy of* $\mathcal{X}$ *given* $\mathcal{Y}$, denoted by $\widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{Y})$, is defined as follows:

$$\widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{Y}) := -\log_2\Big(\mathop{\mathbf{E}}_{y \leftarrow \mathcal{Y}}\Big[\max_x \Pr[\mathcal{X} = x | \mathcal{Y} = y]\Big]\Big).$$

Universal Hash Family. Let D and R be finite sets, and let $\mathcal{H} = \{H : D \to R\}$ be a family of functions. $\mathcal{H}$ is said to be a universal hash family, if for all distinct $x, x' \in D$, we have $\Pr_{H \leftarrow \mathcal{H}}[H(x) = H(x')] \leq |R|^{-1}$.

Leftover Hash Lemma. In this paper, we will use the following version of the leftover hash lemma [HILL99].

Lemma 1 ([DRS04]). *Let D and R be finite sets, and let $\mathcal{H} = \{H : D \to R\}$ be a universal hash family. Let $(\mathcal{X}, \mathcal{Y})$ be a joint distribution such that $\mathcal{X}$ is distributed over D. Then, for all (computationally unbounded) adversaries $\mathcal{A}$, we have*

$$\left|\Pr[\mathcal{A}(\mathcal{D}^{\texttt{LHL-real}}_{\mathcal{H},(\mathcal{X},\mathcal{Y})}) = 1] - \Pr[\mathcal{A}(\mathcal{D}^{\texttt{LHL-ideal}}_{\mathcal{H},(\mathcal{X},\mathcal{Y})}) = 1]\right| \leq \frac{1}{2}\sqrt{|R| \cdot 2^{-\widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{Y})}},$$

where $\mathcal{D}^{\texttt{LHL-real}}_{\mathcal{H},(\mathcal{X},\mathcal{Y})}$ and $\mathcal{D}^{\texttt{LHL-ideal}}_{\mathcal{H},(\mathcal{X},\mathcal{Y})}$ are the distributions defined as follows:

$$\mathcal{D}^{\texttt{LHL-real}}_{\mathcal{H},(\mathcal{X},\mathcal{Y})} := \Big\{H \leftarrow \mathcal{H};\ (x,y) \leftarrow (\mathcal{X},\mathcal{Y});\ K^* \leftarrow H(x) : (H, y, K^*)\Big\},$$
$$\mathcal{D}^{\texttt{LHL-ideal}}_{\mathcal{H},(\mathcal{X},\mathcal{Y})} := \Big\{H \leftarrow \mathcal{H};\ (x,y) \leftarrow (\mathcal{X},\mathcal{Y});\ K^* \leftarrow R : (H, y, K^*)\Big\}.$$

4 Chosen Ciphertext Security via BARGs

In this section, as our main technical result, we show how to construct a wCCA secure TBE scheme from the combination of a CPA secure PKE scheme, a TBEQC scheme, and a BARG.

4.1 Proposed Construction

The proposed TBE scheme uses the following primitives as building blocks:

- Let $\Pi = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme whose plaintext space is $\{0,1\}^\lambda$. For simplicity of notation and ease of understanding, we slightly abuse the syntax and allow Enc to treat the invalidity symbol "$\perp$" as a plaintext.[6] We denote the randomness space of Enc by $\mathcal{R}_\lambda$, the public key size by $\ell_{pk} = \ell_{pk}(\lambda)$, and the ciphertext size by $\ell_c = \ell_c(\lambda)$ when the security parameter is λ.
- Let $\mathcal{C} = (\mathsf{CKG}, \mathsf{Com}, \mathsf{EQSetup})$ be a TBEQC scheme in which the randomness space of Com is $\{0,1\}^\lambda$.[7] We denote the commitment key size by $\ell_{ck} = \ell_{ck}(\lambda)$ and the commitment size by $\ell_{com} = \ell_{com}(\lambda)$ when the security parameter is λ.

[6] This property can be generically achieved from a PKE scheme with plaintext space $\{0,1\}^{\lambda+1}$, by putting a prefix indicator bit, say 1, to an ordinary plaintext in $\{0,1\}^\lambda$, and encoding $\perp$ as $0\|0^\lambda$.

[7] Such a TBEQC scheme can be built from any PRG (and hence from any one-way function). See the proof of Theorem 3.

– Define the NP language $\mathcal{L} = \{\mathcal{L}_\lambda\}_{\lambda\in\mathbb{N}}$ by

$$\mathcal{L}_\lambda := \left\{ (pk^0, pk^1, ck, \mathsf{t}, c^0, c^1, \gamma) \;\middle|\; \begin{array}{l} \exists (s, \rho, r) \in \{0,1\} \times \{0,1\}^\lambda \times \mathcal{R}_\lambda \text{ s.t.} \\ c^s = \mathsf{Enc}(pk^s, \rho; r) \wedge \gamma = \mathsf{Com}(ck, \mathsf{t}, s; \rho) \end{array} \right\},$$

where $pk^0, pk^1 \in \{0,1\}^{\ell_{pk}}$, $ck \in \{0,1\}^{\ell_{ck}}$, $\mathsf{t} \in \{0,1\}^\lambda$, $c^0, c^1 \in \{0,1\}^{\ell_c}$, and $\gamma \in \{0,1\}^{\ell_{com}}$.
Let n $(= \mathsf{poly}(\lambda))$ be a polynomial that denotes the bit-length of each statement in $\mathcal{L}_\lambda$. Namely, $n = n(\lambda) := 2(\ell_{pk} + \ell_c) + \ell_{ck} + \ell_{com} + \lambda$.
– Let $\mathcal{P} = (\mathsf{CRSG}, \mathsf{BProve}, \mathsf{BVerify})$ be a k^ϵ-succinct BARG (with any non-negative constant $\epsilon < 1$) for the above NP language $\mathcal{L}$.
Since $\mathcal{P}$ is k^ϵ-succinct and $n = \mathsf{poly}(\lambda)$, there exists a polynomial $p = p(\lambda)$ such that the size of a proof π generated by BProve when proving the correctness of k statements in $\mathcal{L}_\lambda$ satisfies $|\pi| \leq p \cdot k^\epsilon$.
– Let $\mathcal{H} = \{H : \{0,1\}^\kappa \to \{0,1\}^\lambda\}$ be a universal hash family, where $\kappa = \kappa(\lambda)$ is any polynomial satisfying the following inequality:

$$\kappa \geq \max\left\{ (2p)^{\frac{1}{1-\epsilon}},\ 6\lambda \right\}. \tag{4}$$

Using the above building blocks, the proposed TBE scheme $\mathcal{T} = (\mathsf{TKG}, \mathsf{TEnc}, \mathsf{TDec})$ with a plaintext space $\{0,1\}^\lambda$ is constructed as in Fig. 1. We note that if one wants to encrypt a longer plaintext, one can use the value $H(s)$ computed in TEnc as a key for symmetric-key encryption to encrypt the plaintext. We omit such an extension for simplicity.

The correctness and wCCA security of the TBE scheme $\mathcal{T}$ is guaranteed by the following theorems.

Theorem 4. *If the PKE scheme Π and the BARG $\mathcal{P}$ satisfy correctness, then so does the TBE scheme $\mathcal{T}$.*

Theorem 5. *Assume that the PKE scheme Π is CPA secure, $\mathcal{C}$ is a secure TBEQC scheme, and the BARG $\mathcal{P}$ satisfies semi-adaptive soundness. Then, the TBE scheme $\mathcal{T}$ is wCCA secure.*

The proofs of Theorems 4 and 5 are provided in Sections 4.2 and 4.3, respectively.

As a direct corollary of the above theorems and Theorems 2 and 3, we obtain our main result: a new set of generic primitives whose existence implies the existence of a CCA secure PKE scheme.

Theorem 6. *Let $\epsilon < 1$ be any non-negative constant. If there exist a CPA secure PKE scheme and a k^ϵ-succinct BARG for NP satisfying semi-adaptive soundness, then there exists a CCA secure PKE scheme.*

4.2 Proof of Theorem 4 (Correctness)

Suppose $C = ((c_i^0, c_i^1, \gamma_i)_i, \pi, c_m)$ is output by $\mathsf{TEnc}(PK, \mathsf{t}, m)$, and consider the behavior of $\mathsf{TDec}(SK, \mathsf{t}, C)$. Firstly, the correctness of the BARG $\mathcal{P}$ implies that

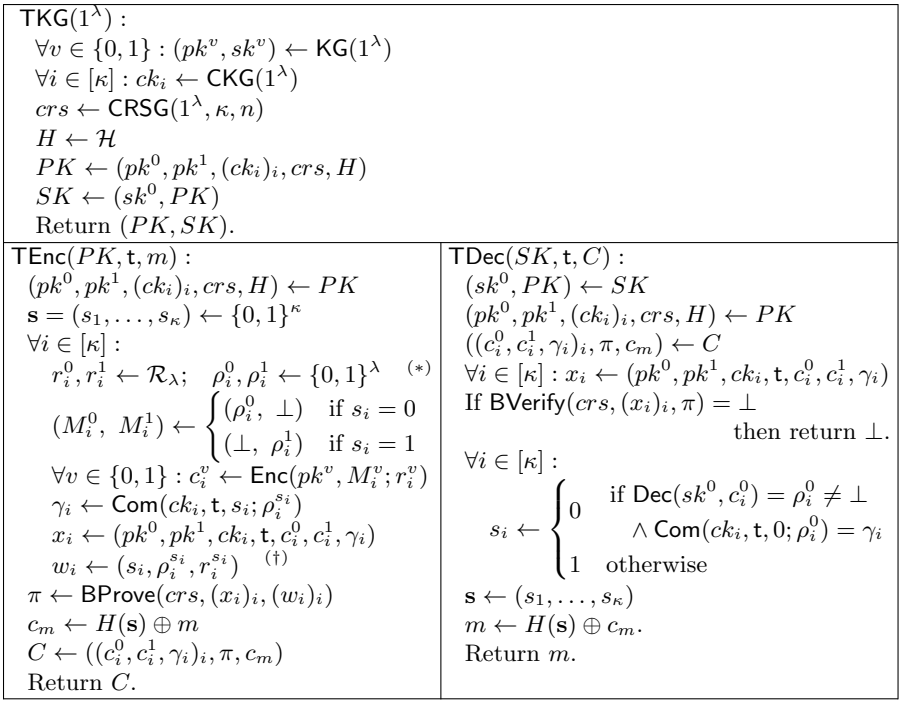

Fig. 1. Construction of a wCCA secure TBE scheme based on a CPA secure PKE scheme, a TBEQC scheme, and a BARG. The notation like $(X_i)_i$ is an abbreviation for $(X_i)_{i\in[\kappa]}$. (*) $\rho_i^{1-s_i}$ is never used in an execution of TEnc, but is included to ease the explanation in the security proof. (†) Each w_i is a witness for $x_i \in \mathcal{L}_\lambda$.

$\mathsf{BVerify}$ in TDec outputs $\top$. Secondly, for the positions $i \in [\kappa]$ for which $s_i = 0$, c_i^0 encrypts the commitment randomness $\rho_i^0 \in \{0,1\}^\lambda$, and γ_i is a commitment of 0 using ρ_i^0, and thus the correctness of the PKE scheme Π implies that TDec recovers $s_i = 0$. Thirdly, for the positions $i \in [\kappa]$ for which $s_i = 1$, c_i^0 encrypts $\perp$, and thus again the correctness of Π implies that TDec recovers $s_i = 1$. Therefore, the $\mathbf{s} = (s_1, \ldots, s_\kappa) \in \{0,1\}^\kappa$ recovered in TDec is exactly the same as the one used in TEnc to generate C, which implies that TDec finally outputs $H(\mathbf{s}) \oplus c_m = m$. □ (**Theorem 4**)

4.3 Proof of Theorem 5 (wCCA Security)

Let $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ be an arbitrary PPT adversary that attacks the wCCA security of the TBE scheme $\mathcal{T}$. We will show that $\mathsf{Adv}^{\mathsf{wCCA}}_{\mathcal{T},\mathcal{A}}(\lambda)$ is negligible.

For this $\mathcal{A}$, we consider a sequence of the following six games.[8]

[8] In the following, we put an asterisk (*) for the values related to the generation of the challenge ciphertext $C^* = ((c_i^{*0}, c_i^{*1}, \gamma_i^*)_i, \pi^*, c_m^*)$. Furthermore, the steps that are not explicitly mentioned are untouched and unchanged from the previous game.

Game 1: This game is $\mathsf{Expt}^{\mathtt{wcca}}_{\mathcal{T},\mathcal{A}}(\lambda)$ itself. In the following, for each $t \in [6]$, let S_t be the event that $\mathcal{A}_2$ succeeds in guessing the challenge bit (i.e. $b' = b$ occurs) in Game t. By definition, we have $\mathsf{Adv}^{\mathtt{wCCA}}_{\mathcal{T},\mathcal{A}}(\lambda) = 2 \cdot |\Pr[\mathsf{S}_1] - 1/2|$.

For completeness, we give a detailed description of Game 1. At the beginning, $(\mathsf{t}^*, \mathsf{st}_0) \leftarrow \mathcal{A}_0(1^\lambda)$ is executed. Next, the key pair (PK, SK) is generated as follows:

1. For each $v \in \{0,1\}$, compute $(pk^v, sk^v) \leftarrow \mathsf{KG}(1^\lambda)$.
2. For each $i \in [\kappa]$, compute $ck_i \leftarrow \mathsf{CKG}(1^\lambda)$.
3. Compute $crs \leftarrow \mathsf{CRSG}(1^\lambda, \kappa, n)$.
4. Pick $H \leftarrow \mathcal{H}$.
5. Set $PK = (pk^0, pk^1, (ck_i)_i, crs, H)$ and $SK = (sk^0, PK)$.

Then, $\mathcal{A}_1(PK, \mathsf{st}_0)$ is executed. At this point, $\mathcal{A}_1$ may start making decryption queries $(\mathsf{t}, C = ((c_i^0, c_i^1, \gamma_i)_i, \pi, c_m))$ with $\mathsf{t} \neq \mathsf{t}^*$. Each of the decryption queries is responded as follows:

1. For each $i \in [\kappa]$, set $x_i = (pk^0, pk^1, ck_i, \mathsf{t}, c_i^0, c_i^1, \gamma_i)$.
2. Run $\mathsf{BVerify}(crs, (x_i)_i, \pi)$. If the result is $\perp$, then return $\perp$ to $\mathcal{A}_1$.
3. For each $i \in [\kappa]$, compute $s_i \in \{0,1\}$ as follows:

$$s_i \leftarrow \begin{cases} 0 & \text{if } \mathsf{Dec}(sk^0, c_i^0) = \rho_i^0 \neq \perp \wedge \mathsf{Com}(ck_i, \mathsf{t}, 0; \rho_i^0) = \gamma_i \\ 1 & \text{otherwise} \end{cases}. \tag{5}$$

4. Set $\mathbf{s} = (s_1, \ldots, s_\kappa) \in \{0,1\}^\kappa$.
5. Return $m = H(\mathbf{s}) \oplus c_m$ to $\mathcal{A}_1$.

At some point, $\mathcal{A}_1$ terminates with output (m_0, m_1, st). Next, the challenge bit $b \leftarrow \{0,1\}$ is picked, and the challenge ciphertext $C^* = ((c_i^{*0}, c_i^{*1}, \gamma_i^*)_i, \pi^*, c_m^*)$ is computed as follows:

1. Pick $\mathbf{s}^* = (s_1^*, \ldots, s_\kappa^*) \leftarrow \{0,1\}^\kappa$.
2. For each $i \in [\kappa]$, do the following:
   (a) Pick $r_i^{*0}, r_i^{*1} \leftarrow \mathcal{R}_\lambda$ and $\rho_i^{*0}, \rho_i^{*1} \leftarrow \{0,1\}^\lambda$.
   (b) Set $(M_i^{*0},\ M_i^{*1}) \leftarrow \begin{cases} (\rho_i^{*0},\ \perp) & \text{if } s_i^* = 0 \\ (\perp,\ \rho_i^{*1}) & \text{if } s_i^* = 1 \end{cases}$.
   (c) For both $v \in \{0,1\}$, compute $c_i^{*v} \leftarrow \mathsf{Enc}(pk^v, M_i^{*v}; r_i^{*v})$.
   (d) Compute $\gamma_i^* \leftarrow \mathsf{Com}(ck_i, \mathsf{t}^*, s_i^*; \rho_i^{*(s_i^*)})$.
   (e) Set $x_i^* = (pk^0, pk^1, ck_i, \mathsf{t}_i^*, c_i^{*0}, c_i^{*1}, \gamma_i^*)$ and $w_i^* = (s_i^*, \rho_i^{*(s_i^*)}, r_i^{*(s_i^*)})$.
3. Compute $\pi^* \leftarrow \mathsf{BProve}(crs, (x_i^*)_i, (w_i^*)_i)$.
4. Compute $c_m^* = H(\mathbf{s}^*) \oplus m_b$.
5. Set $C^* = ((c_i^{*0}, c_i^{*1}, \gamma_i^*)_i, \pi^*, c_m^*)$.

Then, $\mathcal{A}_2(C^*, \mathsf{st})$ is executed. $\mathcal{A}_2$'s decryption queries are responded in the same way as for $\mathcal{A}_1$. Finally, at some point, $\mathcal{A}_2$ terminates with its guess bit b' for b.

Game 2: In this game, we modify some steps for generating PK and C^* so that we use the EQ mode of the underlying TBEQC scheme $\mathcal{C}$. Specifically, for each $i \in [\kappa]$, we generate ck_i, γ_i^*, ρ_i^{*0}, and ρ_i^{*1} by $(ck_i, \gamma_i^*, \rho_i^{*0}, \rho_i^{*1}) \leftarrow \mathsf{EQSetup}(1^\lambda, \mathsf{t}^*)$.

Note that in both Games 1 and 2, $\rho_i^{*(1-s_i^*)}$ is information-theoretically hidden from $\mathcal{A}$'s view. Hence, from $\mathcal{A}$'s viewpoint, the difference between these games lies only in how the tuple $(ck_i, \gamma_i^*, \rho_i^{*(s_i^*)})$ is generated. Furthermore, this tuple is indistinguishable between Games 1 and 2 due to the indistinguishability of the two modes of the underlying TBEQC scheme $\mathcal{C}$. Hence, applying a hybrid argument regarding the index $i \in [\kappa]$ yields that $|\Pr[\mathsf{S}_1] - \Pr[\mathsf{S}_2]|$ is negligible. (For completeness, we provide the reduction in Appendix B.1.)

Game 3: In this game, we modify a part of the steps for generating C^*. Specifically, for each $i \in [\kappa]$, we set the pair (M_i^{*0}, M_i^{*1}) as follows:

$$\left(M_i^{*0},\ M_i^{*1}\right) \leftarrow \begin{cases} \left(\rho_i^{*0},\ \rho_i^{*1}\right) & \text{if } s_i^* = 0 \\ \left(\bot,\ \rho_i^{*1}\right) & \text{if } s_i^* = 1 \end{cases}.$$

By the above modification, in this and subsequent games, we always have $M_i^{*1} = \rho_i^{*1}$. (However, we have not changed how M_i^{*0} is generated.)

In both Games 2 and 3, sk^1 is never used. Furthermore, for the indices $i \in [\kappa]$ at which $s_i^* = 0$ holds, r_i^{*1} is used only to generate c_i^{*1}, and not used as part of the witness w_i^*. Hence, by the CPA security of the underlying PKE scheme Π under pk^1, we have that $|\Pr[\mathsf{S}_2] - \Pr[\mathsf{S}_3]|$ is negligible. (For completeness, we provide the reduction in Appendix B.2.)

Game 4: In this game, we modify the decryption oracle so that it decrypts a queried ciphertext by using sk^1, instead of using sk^0. Specifically, the step for computing each s_i in the decryption oracle is now changed as follows:

$$s_i \leftarrow \begin{cases} 1 & \text{if } \mathsf{Dec}(sk^1, c_i^1) = \rho_i^1 \neq \bot \wedge \mathsf{Com}(ck_i, \mathsf{t}, 1; \rho_i^1) = \gamma_i \\ 0 & \text{otherwise} \end{cases}. \tag{6}$$

We will later show in Lemma 2 that $|\Pr[\mathsf{S}_3] - \Pr[\mathsf{S}_4]|$ is negligible by the statistical binding in the EQ mode of the underlying TBEQC scheme $\mathcal{C}$ and the semi-adaptive soundness of the underlying BARG $\mathcal{P}$.

Game 5: In this game, we again modify a part of the steps for generating C^*. Specifically, for each $i \in [\kappa]$, we set the pair (M_i^{*0}, M_i^{*1}) as follows, now regardless of whether $s_i^* = 0$ or $s_i^* = 1$:

$$\left(M_i^{*0},\ M_i^{*1}\right) \leftarrow \left(\rho_i^{*0},\ \rho_i^{*1}\right).$$

By the above modification, we now always have $M_i^{*0} = \rho_i^{*0}$.

In both Games 4 and 5, sk^0 is never used. Furthermore, for the indices $i \in [\kappa]$ at which $s_i^* = 1$ holds, r_i^{*0} is used only to generate c_i^{*0}, and not used as part of the witness w_i^*. Hence, by the CPA security of the underlying PKE scheme Π

under pk^0, we have that $|\Pr[\mathsf{S}_4] - \Pr[\mathsf{S}_5]|$ is negligible. (The reduction is very similarly to that between Games 2 and 3, and thus omitted.)

Moreover, observe that in Game 5, c^{*v} is an encryption of $M_i^{*v} = \rho_i^{*v}$ for every $(i, v) \in [\kappa] \times \{0, 1\}$. Furthermore, γ_i^* is generated by using $\mathsf{EQSetup}$ in Game 5. These mean that the components $(c_i^{*0}, c_i^{*1}, \gamma_i^*)_i$ in the challenge ciphertext C^* do not depend on the value $\mathbf{s}^* = (s_1^*, \ldots, s_\kappa^*)$. Put differently, in Game 5, the only components of C^* that are dependent on $\mathbf{s}^*$, are the BARG proof π^* and the last component $c_m^* = H(\mathbf{s}^*) \oplus m_b$.

Game 6: In this game, we further modify a part of the steps for generating C^*. Specifically, we pick $K^* \leftarrow \{0,1\}^\lambda$ uniformly at random and compute $c_m^* \leftarrow K^* \oplus m_b$, instead of $c_m^* \leftarrow H(\mathbf{s}^*) \oplus m_b$.

Since the information of the challenge bit b is completely hidden from $\mathcal{A}$'s view in Game 6, we have $\Pr[\mathsf{S}_6] = 1/2$.

We will later show in Lemma 3 that the leftover hash lemma (Lemma 1) implies that $|\Pr[\mathsf{S}_5] - \Pr[\mathsf{S}_6]| \leq 2^{-\lambda-1}$ and thus is negligible.

The above completes the description of the games. To complete the proof, it remains to show that $|\Pr[\mathsf{S}_3] - \Pr[\mathsf{S}_4]|$ and $|\Pr[\mathsf{S}_5] - \Pr[\mathsf{S}_6]|$ are negligible, which we show as Lemmas 2 and 3, respectively, in the following.

Lemma 2. *$|\Pr[\mathsf{S}_3] - \Pr[\mathsf{S}_4]|$ is negligible by the statistical binding in the EQ mode of the underlying TBEQC scheme $\mathcal{C}$ and the semi-adaptive soundness of the underlying BARG $\mathcal{P}$.*

Proof of Lemma 2. For a decryption query $(\mathsf{t}, C = ((c_i^0, c_i^1, \gamma_i)_i, \pi, c_m))$ made by $\mathcal{A}$ that is not rejected by the verification of π, let $\rho_i^0 = \mathsf{Dec}(sk_i^0, c_i^0)$ (resp. $\rho_i^1 = \mathsf{Dec}(sk_i^1, c_i^1)$), and let s_i^0 (resp. s_i^1) be the i-th bit of $\mathbf{s}$ computed as in Eq. (5) (resp. Eq. (6)) for each $i \in [\kappa]$. Furthermore, for each $i \in [\kappa]$, let $x_i = (pk^0, pk^1, ck_i, \mathsf{t}, c_i^0, c_i^1, \gamma_i)$.

Note that Games 3 and 4 behave identically unless $\mathcal{A}$ submits at least one decryption query that simultaneously satisfies the following conditions (since otherwise the answer of the decryption oracle in both games is identical):

- $\mathsf{BVerify}(crs, (x_i)_i, \pi) = \top$.
- There exists an index $j \in [\kappa]$ at which $s_j^0 \neq s_j^1$ holds.

Let us call such a decryption query *critical*, and call the index j satisfying the second condition the *critical index*.[9]

We further classify critical queries. We call a critical query *Type-1* if $(s_j^0, s_j^1) = (0, 1)$ holds at the critical index j, and otherwise (i.e. if $(s_j^0, s_j^1) = (1, 0)$), we call the query *Type-2*. Let us denote the events that $\mathcal{A}$ submits at least one critical (resp, Type-1 critical, Type-2 critical) query in Game $t \in \{3, 4\}$ by Q_t (resp. $\mathsf{Q}_t^{(1)}$, $\mathsf{Q}_t^{(2)}$). Then, by definition, we have

$$|\Pr[\mathsf{S}_3] - \Pr[\mathsf{S}_4]| \leq \Pr[\mathsf{Q}_4] = \Pr[\mathsf{Q}_3] \leq \Pr[\mathsf{Q}_3^{(1)}] + \Pr[\mathsf{Q}_3^{(2)}].$$

[9] If there are multiple indices $j \in [\kappa]$ satisfying the second condition, then we focus on the smallest one among them to make a critical index unique to each critical query.

Below, we show that the probability that $\mathcal{A}$ submits either of the above types of critical queries in Game 3 is negligible.

If $\mathcal{A}$ submits a Type-1 critical query, then by the definitions of Eqs. (5) and (6), the pair ρ_j^0 and ρ_j^1 simultaneously satisfies $\rho_j^0 \neq \bot$, $\rho_j^1 \neq \bot$, and $\gamma_j = \mathsf{Com}(ck_j, \mathsf{t}, 0; \rho_j^0) = \mathsf{Com}(ck_j, \mathsf{t}, 1; \rho_j^1)$. However, by the statistical binding in the EQ mode of the underlying TBEQC scheme $\mathcal{C}$, the probability that such a pair ρ_j^0, ρ_j^1 (and the tag $\mathsf{t} \neq \mathsf{t}^*$) exists is negligible. Hence, the chance that $\mathcal{A}$ can submit a Type-1 critical query, namely $\Pr[\mathsf{Q}_3^{(1)}]$, is negligible.

On the other hand, if $\mathcal{A}$ submits a Type-2 critical query, then Eq. (5) and $s_j^0 = 1$ imply that either of the following conditions (a) and (b) holds:

(a) $\rho_j^0 = \bot$.
(b) $\rho_j^0 \neq \bot \wedge \mathsf{Com}(ck_j, \mathsf{t}, 0; \rho_j^0) \neq \gamma_j$.

Symmetrically, Eq. (6) and $s_j^1 = 0$ imply that either of the following conditions (c) and (d) holds:

(c) $\rho_j^1 = \bot$.
(d) $\rho_j^1 \neq \bot \wedge \mathsf{Com}(ck_j, \mathsf{t}, 1; \rho_j^1) \neq \gamma_j$.

However, the combination of "(a) or (b)" AND "(c) or (d)", implies that there exists no tuple $(s, \rho, r) \in \{0,1\} \times \{0,1\}^\lambda \times \mathcal{R}_\lambda$ that simultaneously satisfies $\mathsf{Enc}(pk^s, \rho; r) = c_j^s$ and $\mathsf{Com}(ck_j, s; \rho) = \gamma_j$, which in turn implies $x_j \notin \mathcal{L}_\lambda$. Hence, if $\mathcal{A}$ submits a Type-2 critical query with non-negligible probability, we can use $\mathcal{A}$ to break the semi-adaptive soundness of the underlying BARG $\mathcal{P}$ with non-negligible probability. More specifically, consider the following PPT adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ against the semi-adaptive soundness of the underlying BARG $\mathcal{P}$:

$\mathcal{B}_1(1^\lambda)$ picks $i^* \in [\kappa]$ uniformly at random, stores i^* to st, and terminates with output $(\kappa, n, i^*, \mathsf{st})$; $\mathcal{B}_2(crs, \mathsf{st})$ generates all the values (except for crs that it received as input) as in Game 3, and simulates Game 3 for $\mathcal{A}$. When $\mathcal{A}_2$ terminates, $\mathcal{B}_2$ checks if $\mathcal{A}$ has made a critical query whose critical index is i^* (which can be checked by $\mathcal{B}_2$ itself using sk^0 and sk^1). If such a decryption query has not been made, then $\mathcal{B}_2$ simply gives up and aborts. Otherwise, $\mathcal{B}_2$ sets $x_i = (pk^0, pk^1, ck_i, \mathsf{t}, c_i^0, c_i^1, \gamma_i)$ for each $i \in [\kappa]$ from the found critical query and the materials used to simulate Game 3, and also extracts the BARG proof π contained in the clear in the query. Finally, $\mathcal{B}_2$ terminates with output $(x_i)_i$ and π.

It is easy to see that $\mathcal{B}$ can simulate Game 3 perfectly for $\mathcal{A}$, and therefore the probability that $\mathcal{A}$ submits a Type-2 critical query is exactly $\Pr[\mathsf{Q}_3^{(2)}]$. Conditioned on that $\mathcal{A}$ has submitted a Type-2 critical query, the probability that its critical index is i^*, is at least $1/\kappa$, since i^* is information-theoretically hidden from $\mathcal{A}$. Therefore, the probability that $\mathcal{B}$ succeeds in breaking the semi-adaptive soundness of $\mathcal{P}$ is at least $(1/\kappa) \cdot \Pr[\mathsf{Q}_3^{(2)}]$. Since $\mathcal{B}$ is PPT and thus this probability must be negligible, we can conclude that $\Pr[\mathsf{Q}_3^{(2)}]$ is negligible.

Putting things together, due to the statistical binding in the EQ mode of the underlying TBEQC scheme $\mathcal{C}$ and the semi-adaptive soundness of the underlying BARG $\mathcal{P}$, $\Pr[\mathsf{Q}_3^{(1)}] + \Pr[\mathsf{Q}_3^{(2)}]$ is bounded to be negligible, which in turn implies that $|\Pr[\mathsf{S}_3] - \Pr[\mathsf{S}_4]|$ is negligible as well. □ (**Lemma 2**)

Lemma 3. $|\Pr[\mathsf{S}_5] - \Pr[\mathsf{S}_6]| \leq 2^{-\lambda-1}$.

Proof of Lemma 3. We will rely on the leftover hash lemma (Lemma 1). As a preparation, consider the values $\mathbf{s}^* \in \{0,1\}^\kappa$ and y generated by the following procedure, which generates the key materials and the components of the challenge ciphertext except for H and c_m^* in the same way as Game 5 and Game 6:

$$
\begin{aligned}
&(\mathsf{t}^*, \mathsf{st}_0) \leftarrow \mathcal{A}_0(1^\lambda)\\
&\forall v \in \{0,1\} : (pk^v, sk^v) \leftarrow \mathsf{KG}(1^\lambda)\\
&\forall i \in [\kappa] : (ck_i, \gamma_i^*, \rho_i^{*0}, \rho_i^{*1}) \leftarrow \mathsf{EQSetup}(1^\lambda, \mathsf{t}^*)\\
&crs \leftarrow \mathsf{CRSG}(1^\lambda, \kappa, n)\\
&\forall (i,v) \in [\kappa] \times \{0,1\} :\\
&\quad r_i^{*v} \leftarrow \mathcal{R}_\lambda\\
&\quad c_i^{*v} \leftarrow \mathsf{Enc}(pk^v, \rho_i^{*v}; r_i^{*v})\\
&\forall i \in [\kappa] : x_i^* \leftarrow (pk^0, pk^1, ck_i, \mathsf{t}^*, c_i^{*0}, c_i^{*1}, \gamma_i^*)\\
&\mathbf{s}^* = (s_1^*, \ldots, s_\kappa^*) \leftarrow \{0,1\}^\kappa\\
&\forall i \in [\kappa] : w_i^* \leftarrow (s_i^*, \rho_i^{*(s_i^*)}, r_i^{*(s_i^*)})\\
&\pi^* \leftarrow \mathsf{BProve}(crs, (x_i^*)_i, (w_i^*)_i)\\
&y \leftarrow (pk^0, pk^1, (ck_i)_i, crs, (c_i^{*0}, c_i^{*1}, \gamma_i^*)_i, \pi^*, sk^1, \mathsf{st}_0)
\end{aligned}
$$

Now, let $\mathcal{S}^*$ (resp. $\mathcal{Y}$) denote the distribution of $\mathbf{s}^*$ (resp. y) generated by the above procedure. Then, $(\mathcal{S}^*, \mathcal{Y})$ forms a joint distribution such that the only component of y that is dependent on $\mathbf{s}^*$ is π^*. Hence, we have $\widetilde{\mathbf{H}}_\infty(\mathcal{S}^*|\mathcal{Y}) \geq \kappa - |\pi^*|$. Furthermore, due to the k^ϵ-succinctness of the underlying BARG $\mathcal{P}$ and the definition of κ (Eq. (4)), we have $|\pi^*| \leq p \cdot \kappa^\epsilon$. Combined together, we have

$$\widetilde{\mathbf{H}}_\infty(\mathcal{S}^*|\mathcal{Y}) \geq \kappa - p \cdot \kappa^\epsilon = \kappa \cdot (1 - p \cdot \kappa^{\epsilon-1}) \overset{(*)}{\geq} \kappa \cdot \frac{1}{2} \overset{(\dagger)}{\geq} 3\lambda, \tag{7}$$

where the inequality $(*)$ uses $\kappa \geq (2p)^{\frac{1}{1-\epsilon}} \Leftrightarrow 1 - p \cdot \kappa^{\epsilon-1} \geq 1/2$, and the inequality $(\dagger)$ uses $\kappa \geq 6\lambda$.

For the above joint distribution $(\mathcal{S}^*, \mathcal{Y})$, we consider an adversary (distinguisher) $\mathcal{B}$ that takes as input (H, y, K^*) that was generated according to one of the two distributions $\mathcal{D}^{\mathtt{LHL\text{-}real}}_{\mathcal{H},(\mathcal{S}^*,\mathcal{Y})}$ and $\mathcal{D}^{\mathtt{LHL\text{-}ideal}}_{\mathcal{H},(\mathcal{S}^*,\mathcal{Y})}$ considered in Lemma 1 (and hence, either K^* is $H(\mathbf{s}^*)$ or completely random), and tries to distinguish them. The description of $\mathcal{B}$ is as follows.

$\mathcal{B}$ receives H, $y = (pk^0, pk^1, (ck_i)_i, crs, (c_i^{*0}, c_i^{*1}, \gamma_i^*)_i, \pi^*, sk^1, \mathsf{st}_0)$, and K^* as input. $\mathcal{B}$ then sets $PK = (pk^0, pk^1, (ck_i)_i, crs, H)$, and runs $\mathcal{A}_1(PK, \mathsf{st}_0)$. For decryption queries (t, C) from $\mathcal{A}_1$, $\mathcal{B}$ responds to each of them using sk^1 in exactly the same way as the decryption oracle in Game 5 does. When $\mathcal{A}_1$ terminates with output (m_0, m_1, st), $\mathcal{B}$ picks $b \leftarrow \{0,1\}$, sets $c_m^* \leftarrow K^* \oplus m_b$ and $C^* \leftarrow ((c_i^{*0}, c_i^{*1}, \gamma_i^*)_i, \pi^*, c_m^*)$. Then, $\mathcal{B}$ runs $\mathcal{A}_2(C^*, \mathsf{st})$, where $\mathcal{A}_2$'s decryption

queries are answered as above. When $\mathcal{A}_2$ terminates with output b', $\mathcal{B}$ sets $\beta' \leftarrow (b' \stackrel{?}{=} b)$ and terminates with output β'.

If $\mathcal{B}$'s input is sampled according to $\mathcal{D}^{\mathtt{LHL\text{-}real}}_{\mathcal{H},(\mathcal{S}^*,\mathcal{Y})}$, then $\mathcal{B}$ simulates Game 5 (where c^*_m is computed as $c^*_m = H(\mathbf{s}^*) \oplus m_b$) perfectly for $\mathcal{A}$. On the other hand, if $\mathcal{B}$'s input is sampled according to $\mathcal{D}^{\mathtt{LHL\text{-}ideal}}_{\mathcal{H},(\mathcal{S}^*,\mathcal{Y})}$, $\mathcal{B}$ simulates Game 6 (where c^*_m is computed as $c^*_m = K^* \oplus m_b$ with a random $K^* \in \{0,1\}^\lambda$) perfectly for $\mathcal{A}$. Since $\mathcal{B}$ outputs 1 only when $b' = b$ occurs, we have $\Pr[\mathcal{B}(\mathcal{D}^{\mathtt{LHL\text{-}real}}_{\mathcal{H},(\mathcal{S}^*,\mathcal{Y})}) = 1] = \Pr[\mathsf{S}_5]$ and $\Pr[\mathcal{B}(\mathcal{D}^{\mathtt{LHL\text{-}ideal}}_{\mathcal{H},(\mathcal{S}^*,\mathcal{Y})}) = 1] = \Pr[\mathsf{S}_6]$.

Then, the leftover hash lemma (Lemma 1) yields the desired inequality as follows:

$$\begin{aligned}\Big|\Pr[\mathsf{S}_5] - \Pr[\mathsf{S}_6]| &= \Big|\Pr[\mathcal{B}(\mathcal{D}^{\mathtt{LHL\text{-}real}}_{\mathcal{H},(\mathcal{S}^*,\mathcal{Y})}) = 1] - \Pr[\mathcal{B}(\mathcal{D}^{\mathtt{LHL\text{-}ideal}}_{\mathcal{H},(\mathcal{S}^*,\mathcal{Y})}) = 1]\Big| \\ &\leq \frac{1}{2} \cdot \sqrt{2^\lambda \cdot 2^{-\widetilde{\mathbf{H}}_\infty(\mathcal{S}^*|\mathcal{Y})}} \stackrel{(*)}{\leq} 2^{-\lambda-1},\end{aligned}$$

where the inequality (*) uses Eq. (7). □ (**Lemma 3**)

As we have seen, for every $t \in [5]$, we have that $|\Pr[\mathsf{S}_t] - \Pr[\mathsf{S}_{t+1}]|$ is negligible and that $|\Pr[\mathsf{S}_6] - 1/2| = 0$. This, together with the triangle inequality, implies that $\mathsf{Adv}^{\mathtt{wCCA}}_{\mathcal{T},\mathcal{A}}(\lambda) = 2 \cdot |\Pr[\mathsf{S}_1] - 1/2|$ is negligible as well. Hence, we can conclude that $\mathcal{T}$ is wCCA secure. □ (**Theorem 5**)

Remark on Privately-Verifiable BARGs. Our proposed construction uses a BARG that is publicly verifiable, meaning that to verify a BARG proof π, the verification algorithm $\mathsf{BVerify}$ need not use any secret state (called a verification key) that cannot be made public to satisfy its semi-adaptive soundness. Note that any publicly-verifiable argument/proof system is also a privately-verifiable one by considering an empty verification key, while the converse is not true in general, and thus the latter is potentially easier to achieve/construct than the former. Thus, it is natural to ask whether we can use a privately-verifiable BARG in our proposed construction.

It is not hard to observe that even if we replace the publicly-verifiable BARG in our proposed TBE construction with a privately-verifiable one so that the verification key (generated together with crs) is included as part of a secret key for the constructed TBE scheme, then the security proof for such modified construction still goes through, as long as the underlying privately-verifiable BARG satisfies a *reusable* variant of semi-adaptive soundness, in the security experiment of which an adversary (cheating prover) is given access to the verification oracle (as many times as it wants) but is not given the secret verification key itself.[10] To see this, note that the only place we rely on the semi-adaptive soundness of the underlying BARG in the proof of Theorem 5 is in the proof of Lemma 2, and the reduction algorithm $\mathcal{B}$ attacking semi-adaptive soundness of

[10] For completeness, we give the formal definition for a privately-verifiable BARG and its reusable semi-adaptive soundness in Appendix C.

the underlying BARG considered there can work (without having the secret key verification by itself) as long as it has access to the verification oracle. Therefore, a privately-verifiable BARG satisfying reusable semi-adaptive soundness suffices for constructing a wCCA secure TBE scheme, and also a CCA secure PKE scheme.

Note that any publicly-verifiable BARG with semi-adaptive soundness is also a privately-verifiable one with reusable semi-adaptive soundness, since simulating the verification oracle is trivial in the publicly verifiable setting. At the moment we do not know whether a privately-verifiable BARG with reusable semi-adaptive soundness is easier to construct than a publicly-verifiable BARG, and we would like to leave it as an interesting open problem.

References

[BDPR98] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 26–45. Springer, Heidelberg, August 1998.

[BGI+01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 1–18. Springer, Heidelberg, August 2001.

[BHK13] Mihir Bellare, Viet Tung Hoang, and Sriram Keelveedhi. Instantiating random oracles via UCEs. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 398–415. Springer, Heidelberg, August 2013.

[BHY09] Mihir Bellare, Dennis Hofheinz, and Scott Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 1–35. Springer, Heidelberg, April 2009.

[BKP+23] Nir Bitansky, Chethan Kamath, Omer Paneth, Ron Rothblum, and Prashant Nalini Vasudevan. Batch proofs are statistically hiding. Cryptology ePrint Archive, Report 2023/754, 2023. `https://eprint.iacr.org/2023/754`.

[Ble98] Daniel Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 1–12. Springer, Heidelberg, August 1998.

[BP04] Mihir Bellare and Adriana Palacio. Towards plaintext-aware public-key encryption without random oracles. In Pil Joong Lee, editor, *ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 48–62. Springer, Heidelberg, December 2004.

[BRS03] John Black, Phillip Rogaway, and Thomas Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In Kaisa Nyberg and Howard M. Heys, editors, *SAC 2002*, volume 2595 of *LNCS*, pages 62–75. Springer, Heidelberg, August 2003.

[CHK04] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–222. Springer, Heidelberg, May 2004.

[CJJ21] Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. Non-interactive batch arguments for NP from standard assumptions. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 394–423, Virtual Event, August 2021. Springer, Heidelberg.

[CJJ22] Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. SNARGs for $\mathcal{P}$ from LWE. In *62nd FOCS*, pages 68–79. IEEE Computer Society Press, February 2022.

[CW23] Jeffrey Champion and David J. Wu. Non-interactive zero-knowledge from non-interactive batch arguments. CRYPTO 2023. Full version is in Cryptology ePrint Archive, Report 2023/695, 2023. `https://eprint.iacr.org/2023/695`.

[Dac14] Dana Dachman-Soled. A black-box construction of a CCA2 encryption scheme from a plaintext aware (sPA1) encryption scheme. In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 37–55. Springer, Heidelberg, March 2014.

[DDN91] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography (extended abstract). In *23rd ACM STOC*, pages 542–552. ACM Press, May 1991.

[DIO98] Giovanni Di Crescenzo, Yuval Ishai, and Rafail Ostrovsky. Non-interactive and non-malleable commitment. In *30th ACM STOC*, pages 141–150. ACM Press, May 1998.

[DRS04] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 523–540. Springer, Heidelberg, May 2004.

[GM82] Shafi Goldwasser and Silvio Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *14th ACM STOC*, pages 365–377. ACM Press, May 1982.

[GMM07] Yael Gertner, Tal Malkin, and Steven Myers. Towards a separation of semantic and CCA security for public key encryption. In Salil P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 434–455. Springer, Heidelberg, February 2007.

[GW11] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 99–108. ACM Press, June 2011.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[HJKS22] James Hulett, Ruta Jawale, Dakshita Khurana, and Akshayaram Srinivasan. SNARGs for P from sub-exponential DDH and QR. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 520–549. Springer, Heidelberg, May / June 2022.

[HK15] Mohammad Hajiabadi and Bruce M. Kapron. Reproducible circularly-secure bit encryption: Applications and realizations. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 224–243. Springer, Heidelberg, August 2015.

[HKW20] Susan Hohenberger, Venkata Koppula, and Brent Waters. Chosen ciphertext security from injective trapdoor functions. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part I*, volume 12170 of *LNCS*, pages 836–866. Springer, Heidelberg, August 2020.

[Kil06] Eike Kiltz. Chosen-ciphertext security from tag-based encryption. In Shai Halevi and Tal Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 581–600. Springer, Heidelberg, March 2006.

[KLVW23] Yael Kalai, Alex Lombardi, Vinod Vaikuntanathan, and Daniel Wichs. Boosting batch arguments and RAM delegation. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 1545–1552. ACM, 2023.

[KM19] Fuyuki Kitagawa and Takahiro Matsuda. CPA-to-CCA transformation for KDM security. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019, Part II*, volume 11892 of *LNCS*, pages 118–148. Springer, Heidelberg, December 2019.

[KM20] Fuyuki Kitagawa and Takahiro Matsuda. Circular security is complete for KDM security. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part I*, volume 12491 of *LNCS*, pages 253–285. Springer, Heidelberg, December 2020.

[KMO10] Eike Kiltz, Payman Mohassel, and Adam O'Neill. Adaptive trapdoor functions and chosen-ciphertext security. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 673–692. Springer, Heidelberg, May / June 2010.

[KMT19] Fuyuki Kitagawa, Takahiro Matsuda, and Keisuke Tanaka. CCA security and trapdoor functions via key-dependent-message security. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 33–64. Springer, Heidelberg, August 2019.

[KPY19] Yael Tauman Kalai, Omer Paneth, and Lisa Yang. How to delegate computations publicly. In Moses Charikar and Edith Cohen, editors, *51st ACM STOC*, pages 1115–1124. ACM Press, June 2019.

[KVZ21] Yael Tauman Kalai, Vinod Vaikuntanathan, and Rachel Yun Zhang. Somewhere statistical soundness, post-quantum security, and SNARGs. In Kobbi Nissim and Brent Waters, editors, *TCC 2021, Part I*, volume 13042 of *LNCS*, pages 330–368. Springer, Heidelberg, November 2021.

[KW19] Venkata Koppula and Brent Waters. Realizing chosen ciphertext security generically in attribute-based encryption and predicate encryption. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 671–700. Springer, Heidelberg, August 2019.

[MH14a] Takahiro Matsuda and Goichiro Hanaoka. Chosen ciphertext security via point obfuscation. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 95–120. Springer, Heidelberg, February 2014.

[MH14b] Takahiro Matsuda and Goichiro Hanaoka. Chosen ciphertext security via UCE. In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 56–76. Springer, Heidelberg, March 2014.

[MH15] Takahiro Matsuda and Goichiro Hanaoka. Constructing and understanding chosen ciphertext security via puncturable key encapsulation mechanisms. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part I*, volume 9014 of *LNCS*, pages 561–590. Springer, Heidelberg, March 2015.

[MH16] Takahiro Matsuda and Goichiro Hanaoka. Trading plaintext-awareness for simulatability to achieve chosen ciphertext security. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *PKC 2016, Part I*, volume 9614 of *LNCS*, pages 3–34. Springer, Heidelberg, March 2016.

[MSs12] Steven Myers, Mona Sergi, and abhi shelat. Blackbox construction of a more than non-malleable CCA1 encryption scheme from plaintext awareness. In Ivan Visconti and Roberto De Prisco, editors, *SCN 12*, volume 7485 of *LNCS*, pages 149–165. Springer, Heidelberg, September 2012.

[MY10] Petros Mol and Scott Yilek. Chosen-ciphertext security from slightly lossy trapdoor functions. In Phong Q. Nguyen and David Pointcheval, editors, *PKC 2010*, volume 6056 of *LNCS*, pages 296–311. Springer, Heidelberg, May 2010.

[Nao91] Moni Naor. Bit commitment using pseudorandomness. *Journal of Cryptology*, 4(2):151–158, January 1991.

[NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd ACM STOC*, pages 427–437. ACM Press, May 1990.

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 187–196. ACM Press, May 2008.

[RS92] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 433–444. Springer, Heidelberg, August 1992.

[RS09] Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 419–436. Springer, Heidelberg, March 2009.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In David B. Shmoys, editor, *46th ACM STOC*, pages 475–484. ACM Press, May / June 2014.

[Wee10] Hoeteck Wee. Efficient chosen-ciphertext security via extractable hash proofs. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 314–332. Springer, Heidelberg, August 2010.

[WW22] Brent Waters and David J. Wu. Batch arguments for sfNP and more from standard bilinear group assumptions. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 433–463. Springer, Heidelberg, August 2022.

[YYHK16] Takashi Yamakawa, Shota Yamada, Goichiro Hanaoka, and Noboru Kunihiro. Adversary-dependent lossy trapdoor function from hardness of factoring semi-smooth RSA subgroup moduli. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 3–32. Springer, Heidelberg, August 2016.

A Constructing TBEQC from PRG

Let us first quickly recall the formal definition of a PRG.

Definition 6. *Let $\ell = \ell(\lambda) > \lambda$, and let $\mathsf{G} : \{0,1\}^* \to \{0,1\}^*$ be an efficiently computable function such that $|\mathsf{G}(x)| = \ell(|x|)$ for all $x \in \{0,1\}^*$. We say that G is a secure pseudorandom generator (PRG), if for all PPT adversaries $\mathcal{A}$, the following difference is negligible:*

$$\Big|\Pr_{x\leftarrow\{0,1\}^\lambda}[\mathcal{A}(1^\lambda, \mathsf{G}(x)) = 1] - \Pr_{y\leftarrow\{0,1\}^\ell}[\mathcal{A}(1^\lambda, y) = 1]\Big|.$$

Proof of Theorem 3. Here, we show a construction of a TBEQC scheme from a PRG, which is sufficient for the proof of Theorem 3 due to the connection between a PRG and a one-way function [HILL99].

Let $\mathsf{G} : \{0,1\}^\lambda \to \{0,1\}^{4\lambda}$ be a PRG. Using G as a building block, we construct a TBEQC scheme $\mathcal{C} = (\mathsf{CKG}, \mathsf{Com}, \mathsf{EQSetup})$ in which the randomness

$\mathsf{CKG}(1^\lambda)$:	$\mathsf{Com}(ck, \mathsf{t}, s \in \{0,1\})$:	$\mathsf{EQSetup}(1^\lambda, \mathsf{t}^*)$:
$\mathsf{A}, \mathsf{B} \leftarrow \{0,1\}^{4\lambda}$	$(\mathsf{A}, \mathsf{B}) \leftarrow ck$	$\mathsf{B} \leftarrow \{0,1\}^{4\lambda}$
$ck \leftarrow (\mathsf{A}, \mathsf{B})$	$\rho \leftarrow \{0,1\}^\lambda$	$\widetilde{\rho}_0, \widetilde{\rho}_1 \leftarrow \{0,1\}^\lambda$
Return ck.	$\gamma \leftarrow \mathsf{G}(\rho) + s \cdot (\mathsf{A} + \mathsf{B} \cdot \mathsf{t})$	$\mathsf{A} \leftarrow \mathsf{G}(\widetilde{\rho}_0) - \mathsf{G}(\widetilde{\rho}_1) - \mathsf{B} \cdot \mathsf{t}^*$
	$= \begin{cases} \mathsf{G}(\rho) & \text{if } s = 0 \\ \mathsf{G}(\rho) + \mathsf{A} + \mathsf{B} \cdot \mathsf{t} & \text{if } s = 1 \end{cases}$	$\widetilde{ck} \leftarrow (\mathsf{A}, \mathsf{B})$
	Return γ.	$\widetilde{\gamma} \leftarrow \mathsf{G}(\widetilde{\rho}_0)$
		Return $(\widetilde{ck}, \widetilde{\gamma}, \widetilde{\rho}_0, \widetilde{\rho}_1)$.

Fig. 2. Construction of a TBEQC scheme based on a PRG. In the figure, arithmetic is over $GF(2^{4\lambda})$, and the tags are mapped to $GF(2^{4\lambda})$ by some injection.

space of Com is $\{0,1\}^\lambda$, as described in Fig. 2.[11] In the figure, we identify $\{0,1\}^{4\lambda}$ with $GF(2^{4\lambda})$ so that arithmetic is over $GF(2^{4\lambda})$. Futhermore, we implicitly assume that tags $\mathsf{t} \in \{0,1\}^\lambda$ are mapped to $GF(2^{4\lambda})$ by some injection.

In the following, we show that $\mathcal{C}$ satisfies the two security properties of a TBEQC scheme.

Indistinguishability of the Two Modes. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an arbitrary PPT adversary. For $\mathcal{A}$, we will consider a sequence of four games below where the first (resp. last) game is exactly $\mathsf{Expt}^{\mathtt{normal}}_{\mathcal{C},\mathcal{A}}(\lambda)$ (resp. $\mathsf{Expt}^{\mathtt{eq}}_{\mathcal{C},\mathcal{A}}(\lambda)$), and show that the probability that $\mathcal{A}_2$ finally outputs 1 in the first game is negligibly close to that in the last game.[12]

Game 1: This game is $\mathsf{Expt}^{\mathtt{normal}}_{\mathcal{C},\mathcal{A}}(\lambda)$ itself. In the following, for each $t \in [4]$, let X_t be the event that $\mathcal{A}_2$ outputs 1 in Game t. By definition, we have $\Pr[\mathsf{X}_1] = \Pr[\mathsf{Expt}^{\mathtt{normal}}_{\mathcal{C},\mathcal{A}}(\lambda) = 1]$.

For completeness, we describe the details of Game 1.

1. Run $(\mathsf{t}^*, s, \mathsf{st}) \leftarrow \mathcal{A}_1(1^\lambda)$.
2. Pick $\mathsf{A}, \mathsf{B} \leftarrow \{0,1\}^{4\lambda}$, and set $ck = (\mathsf{A}, \mathsf{B})$.
3. Pick $\rho \leftarrow \{0,1\}^\lambda$, and compute $\gamma \leftarrow \mathsf{G}(\rho) + s \cdot (\mathsf{A} + \mathsf{B} \cdot \mathsf{t}^*)$.
4. Run $b' \leftarrow \mathcal{A}_2(ck, \gamma, \rho)$.

Game 2: In this game, we pick $\widetilde{\rho}_0, \widetilde{\rho}_1 \in \{0,1\}^\lambda$ uniformly at random, and then generate γ as follows:

$$\gamma \leftarrow \mathsf{G}(\widetilde{\rho}_s) + s \cdot (\mathsf{A} + \mathsf{B} \cdot \mathsf{t}^*).$$

Furthermore, we replace ρ in $\mathcal{A}_2$'s input in Game 1 with $\widetilde{\rho}_s$.

Note that the distribution of $\mathcal{A}_2$'s input in Game 2 is identical to that in Game 1. Hence, we have $\Pr[\mathsf{X}_2] = \Pr[\mathsf{X}_1]$.

[11] If multiple commitment keys are setup, then the value B can be shared among all the keys. We do not introduce such a variant for simplicity of exposition.

[12] In the following, the steps that are not explicitly mentioned are untouched and unchanged from the previous game.

Game 3: In this game, we pick $R \in \{0,1\}^{4\lambda}$ uniformly at random, and then generate A as follows:

$$\mathsf{A} \leftarrow \begin{cases} \mathsf{G}(\widetilde{\rho}_0) - R - \mathsf{B} \cdot \mathsf{t}^* & \text{if } s = 0 \\ R - \mathsf{G}(\widetilde{\rho}_1) - \mathsf{B} \cdot \mathsf{t}^* & \text{if } s = 1 \end{cases}.$$

Note that in this game, regardless of whether $s = 0$ or $s = 1$, A is distributed uniformly and independently of any other values due to the mask by R. Hence, the distribution of $\mathcal{A}_2$'s input in Game 3 is identical to that in Game 2, and we have $\Pr[\mathsf{X}_3] = \Pr[\mathsf{X}_2]$.

Game 4: In this game, we generate R by $R \leftarrow \mathsf{G}(\widetilde{\rho}_{1-s})$.

Note that in Game 3, $\widetilde{\rho}_{1-s}$ is information-theoretically hidden from $\mathcal{A}$'s view. Hence, by the security of the PRG G, $|\Pr[\mathsf{X}_4] - \Pr[\mathsf{X}_3]|$ is negligible.

Finally, note that Game 4 is identical to $\mathsf{Expt}^{\mathsf{eq}}_{\mathcal{C},\mathcal{A}}(\lambda)$. Indeed, in Game 4, A is generated as $\mathsf{A} \leftarrow \mathsf{G}(\widetilde{\rho}_0) - \mathsf{G}(\widetilde{\rho}_1) - \mathsf{B} \cdot \mathsf{t}^*$ regardless of whether $s = 0$ or $s = 1$. Furthermore, if $s = 0$ then $\gamma = \mathsf{G}(\widetilde{\rho}_0)$ by definition. That $\gamma = \mathsf{G}(\widetilde{\rho}_0)$ holds even if $s = 1$ can be checked as follows:

$$\begin{aligned} \gamma &= \mathsf{G}(\widetilde{\rho}_1) + \mathsf{A} + \mathsf{B} \cdot \mathsf{t}^* \\ &= \mathsf{G}(\widetilde{\rho}_1) + \left(\mathsf{G}(\widetilde{\rho}_0) - \mathsf{G}(\widetilde{\rho}_1) - \mathsf{B} \cdot \mathsf{t}^*\right) + \mathsf{B} \cdot \mathsf{t}^* \\ &= \mathsf{G}(\widetilde{\rho}_0) \end{aligned}$$

Therefore, we have $\Pr[\mathsf{X}_4] = \Pr[\mathsf{Expt}^{\mathsf{eq}}_{\mathcal{C},\mathcal{A}}(\lambda) = 1]$.

By the above arguments and the triangle inequality, we can conclude that $|\Pr[\mathsf{Expt}^{\mathtt{normal}}_{\mathcal{C},\mathcal{A}}(\lambda) = 1] - \Pr[\mathsf{Expt}^{\mathsf{eq}}_{\mathcal{C},\mathcal{A}}(\lambda) = 1]| = |\Pr[\mathsf{X}_1] - \Pr[\mathsf{X}_4]|$ is negligible. Hence, $\mathcal{C}$ satisfies the indistinguishability of the two modes.

Statistical Binding in the EQ Mode. Fix arbitrarily a tag $\mathsf{t}^* \in \{0,1\}^\lambda$ and the values $\widetilde{\rho}_0, \widetilde{\rho}_1 \in \{0,1\}^\lambda$ generated in an execution of $\mathsf{EQSetup}(1^\lambda, \mathsf{t}^*)$. Using t^*, $\widetilde{\rho}_0$, and $\widetilde{\rho}_1$, we define the function $f_{\mathsf{t}^*,\widetilde{\rho}_0,\widetilde{\rho}_1} : (\{0,1\}^\lambda \setminus \{\mathsf{t}^*\}) \times \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^{4\lambda}$ by

$$f_{\mathsf{t}^*,\widetilde{\rho}_0,\widetilde{\rho}_1}(\mathsf{t}, \rho_0, \rho_1) := \frac{\mathsf{G}(\rho_0) - \mathsf{G}(\rho_1) - \mathsf{G}(\widetilde{\rho}_0) + \mathsf{G}(\widetilde{\rho}_1)}{\mathsf{t} - \mathsf{t}^*}.$$

Now, let us call $\mathsf{B} \in \{0,1\}^{4\lambda}$ *bad* if it belongs to the image of $f_{\mathsf{t}^*,\widetilde{\rho}_0,\widetilde{\rho}_1}$. Otherwise, we call B *good*. Note that the size of the image of $f_{\mathsf{t}^*,\widetilde{\rho}_0,\widetilde{\rho}_1}$ is at most $2^{3\lambda}$. Hence, when $\mathsf{EQSetup}(1^\lambda, \mathsf{t}^*)$ is executed in which $\mathsf{B} \in \{0,1\}^{4\lambda}$ is chosen uniformly at random, the probability that B is bad is at most $2^{3\lambda}/2^{4\lambda} = 2^{-\lambda}$ and hence negligible.

In the following, we show that in case $\widetilde{ck} = (\mathsf{A}, \mathsf{B})$ with a good B is generated by an execution of $\mathsf{EQSetup}(1^\lambda, \mathsf{t}^*)$, there exists no tuple $(\mathsf{t}, \rho_0, \rho_1) \in (\{0,1\}^\lambda \setminus \{\mathsf{t}^*\}) \times \{0,1\}^\lambda \times \{0,1\}^\lambda$ satisfying $\mathsf{Com}(\widetilde{ck}, \mathsf{t}, 0; \rho_0) = \mathsf{Com}(\widetilde{ck}, \mathsf{t}, 1; \rho_1)$.

Due to the condition that B is good, for any $\rho_0, \rho_1 \in \{0,1\}^\lambda$ and $\mathsf{t} \in \{0,1\}^\lambda \setminus \{\mathsf{t}^*\}$, we have

$$\begin{aligned}
& \mathsf{B} \neq f_{\mathsf{t}^*,\widetilde{\rho}_0,\widetilde{\rho}_1}(\mathsf{t},\rho_0,\rho_1) = \frac{\mathsf{G}(\rho_0)-\mathsf{G}(\rho_1)-\mathsf{G}(\widetilde{\rho}_0)+\mathsf{G}(\widetilde{\rho}_1)}{\mathsf{t}-\mathsf{t}^*} \\
\iff\ & \mathsf{G}(\rho_0) \neq \mathsf{G}(\rho_1) + \Big(\mathsf{G}(\widetilde{\rho}_0)-\mathsf{G}(\widetilde{\rho}_1)-\mathsf{B}\cdot\mathsf{t}^*\Big) + \mathsf{B}\cdot\mathsf{t} \\
& \qquad = \mathsf{G}(\rho_1) + \mathsf{A} + \mathsf{B}\cdot\mathsf{t} \\
\iff\ & \mathsf{Com}(\widetilde{ck},\mathsf{t},0;\rho_0) \neq \mathsf{Com}(\widetilde{ck},\mathsf{t},1;\rho_1).
\end{aligned}$$

Hence, for any t^*, unless $\widetilde{ck}$ contains a bad B, no commitment generated under $\widetilde{ck}$ and $\mathsf{t} \neq \mathsf{t}^*$ can be opened to both of 0 and 1. Furthermore, as seen above, the probability that $\widetilde{ck}$ with a bad B is generated in an execution of $\mathsf{EQSetup}(1^\lambda,\mathsf{t}^*)$ is negligible. Hence, $\mathcal{C}$ satisfies statistical binding in the EQ mode. □ (**Theorem 3**)

B Omitted Proofs

B.1 Game 1 vs. Game 2

Here, we show that $|\Pr[\mathsf{S}_1]-\Pr[\mathsf{S}_2]|$ is negligible due to the indistinguishability of the two modes of the underlying TBEQC scheme $\mathcal{C}$.

Let $\mathcal{A}=(\mathcal{A}_0,\mathcal{A}_1,\mathcal{A}_2)$ be the wCCA adversary considered in the proof of Theorem 5. Using $\mathcal{A}$ as a building block, we construct a reduction algorithm $\mathcal{B}=(\mathcal{B}_1,\mathcal{B}_2)$ attacking the indistinguishability of the two modes of the underlying TBEQC scheme $\mathcal{C}$.

$\mathcal{B}_1(1^\lambda)$**:** $\mathcal{B}_1$ initially runs $(\mathsf{t}^*,\mathsf{st}_0)\leftarrow\mathcal{A}_0(1^\lambda)$. Then, $\mathcal{B}_1$ picks $\mathbf{s}^*=(s_1^*,\ldots,s_\kappa^*)\in\{0,1\}^\kappa$ and $u\in[\kappa]$ uniformly at random. Finally, $\mathcal{B}_1$ sets $\mathsf{st}_\mathcal{B}$ as all the values known to $\mathcal{B}_1$, and terminates with output $(\mathsf{t}^*,s_u^*,\mathsf{st}_\mathcal{B})$.

$\mathcal{B}_2(ck',\gamma',\rho',\mathsf{st}_\mathcal{B})$**:** $\mathcal{B}_2$ generates the key materials and the components of the challenge ciphertexts C^* (except for c_m^*) as follows[13]:

1. For each $v\in\{0,1\}$, compute $(pk^v,sk^v)\leftarrow\mathsf{KG}(1^\lambda)$.
2. For each $i\in[\kappa]$, generate $(ck_i,\gamma_i^*,\rho_i^{*(s_i^*)})$ as follows:
   - If $i<u$, compute $(\widetilde{ck}_i,\widetilde{\gamma}_i,\widetilde{\rho}_i^0,\widetilde{\rho}_i^1)\leftarrow\mathsf{EQSetup}(1^\lambda,\mathsf{t}^*)$, and set $(ck_i,\gamma_i^*,\rho_i^{*(s_i^*)})\leftarrow(\widetilde{ck}_i,\widetilde{\gamma}_i,\widetilde{\rho}_i^{(s_i^*)})$.
   - If $i=u$, set $(ck_u,\gamma_u^*,\rho_u^{*(s_u^*)})\leftarrow(ck',\gamma',\rho')$.
   - If $i>u$, pick $\rho_i^{*(s_i^*)}\leftarrow\{0,1\}^\lambda$, and then compute $ck_i\leftarrow\mathsf{CKG}(1^\lambda)$ and $\gamma_i^*\leftarrow\mathsf{Com}(ck_i,\mathsf{t}^*,s_i^*;\rho_i^{*(s_i^*)})$.
3. Compute $crs\leftarrow\mathsf{CRSG}(1^\lambda,\kappa,n)$.
4. Pick $H\leftarrow\mathcal{H}$.

[13] Note that the values $\{\rho_i^{(1-s_i^*)}\}_{i\in[\kappa]}$ are not at all used in Games 1 and 2, and thus $\mathcal{B}_2$ here do not generate them.

5. Set $PK = (pk^0, pk^1, (ck_i)_i, crs, H)$ and $SK = (sk^0, PK)$.
6. For each $i \in [\kappa]$, do the following:
   (a) Pick $r_i^{*0}, r_i^{*1} \leftarrow \mathcal{R}_\lambda$.
   (b) Set $(M_i^{*0},\ M_i^{*1}) \leftarrow \begin{cases} (\rho_i^{*0},\ \bot) & \text{if } s_i^* = 0 \\ (\bot,\ \rho_i^{*1}) & \text{if } s_i^* = 1 \end{cases}$.
   (c) For both $v \in \{0,1\}$, compute $c_i^{*v} \leftarrow \mathsf{Enc}(pk^v, M_i^{*v}; r_i^{*v})$.
   (d) Set $x_i^* = (pk^0, pk^1, ck_i, \mathsf{t}_i^*, c_i^{*0}, c_i^{*1}, \gamma_i^*)$ and $w_i^* = (s_i^*, \rho_i^{*(s_i^*)}, r_i^{*(s_i^*)})$.
7. Compute $\pi^* \leftarrow \mathsf{BProve}(crs, (x_i^*)_i, (w_i^*)_i)$.

Then, $\mathcal{B}_2$ runs $(m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}_1(PK, \mathsf{st}_0)$, where $\mathcal{B}_2$ answers $\mathcal{A}_1$'s decryption queries using SK. $\mathcal{B}_2$ picks the challenge bit $b \leftarrow \{0,1\}$, computes $c_m^* \leftarrow H(\mathsf{s}^*) \oplus m_b$, and sets $C^* = ((c_i^{*0}, c_i^{*1}, \gamma_i^*)_i, \pi^*, c_m^*)$. $\mathcal{B}_2$ then runs $b' \leftarrow \mathcal{A}_2(C^*, \mathsf{st})$, where $\mathcal{A}_2$'s decryption queries are responded as before. Finally, at some point, $\mathcal{A}_2$ terminates with its guess bit b' for b. $\mathcal{B}_2$ sets $\beta' \leftarrow (b' \stackrel{?}{=} b)$, and terminates with output β'.

The above completes the description of $\mathcal{B}$. It is not hard to see the following facts:

- If $\mathcal{B}$ runs in $\mathsf{Expt}^{\mathtt{normal}}_{\mathcal{C},\mathcal{B}}(\lambda)$ and the value u picked by $\mathcal{B}_1$ at the beginning is $u = 1$, then $\mathcal{B}$ simulates Game 1 perfectly for $\mathcal{A}$. Therefore, the probability that $\mathcal{B}$ outputs $\beta' = 1$ is equivalent to the probability that $\mathcal{A}$ succeeds in Game 1, namely, $\Pr[\mathsf{Expt}^{\mathtt{normal}}_{\mathcal{C},\mathcal{B}}(\lambda) = 1 | u = 1] = \Pr[\mathsf{S}_1]$.
- If $\mathcal{B}$ runs in $\mathsf{Expt}^{\mathtt{eq}}_{\mathcal{C},\mathcal{B}}(\lambda)$ and the value u picked by $\mathcal{B}_1$ at the begining is $u = \kappa$, then $\mathcal{B}$ simulates Game 2 perfectly for $\mathcal{A}$. Therefore, the probability that $\mathcal{B}$ outputs $\beta' = 1$ is equivalent to the probability that $\mathcal{A}$ succeeds in Game 2, namely, $\Pr[\mathsf{Expt}^{\mathtt{eq}}_{\mathcal{C},\mathcal{B}}(\lambda) = 1 | u = \kappa] = \Pr[\mathsf{S}_2]$.
- From $\mathcal{A}$'s view point, for any $j \in \{1, \ldots, \kappa - 1\}$, the experiment simulated by $\mathcal{B}$ in the following situations is identically distributed:
  - $\mathcal{B}$ runs in $\mathsf{Expt}^{\mathtt{normal}}_{\mathcal{C},\mathcal{B}}(\lambda)$ and the value u picked by $\mathcal{B}_1$ at the beginning is $u = j + 1$.
  - $\mathcal{B}$ runs in $\mathsf{Expt}^{\mathtt{eq}}_{\mathcal{C},\mathcal{B}}(\lambda)$ and the value u picked by $\mathcal{B}_1$ at the beginning is $u = j$.

  Hence, in these situations, the probability that $\mathcal{A}$ succeeds in guessing the challenge bit, and consequently $\mathcal{B}$ outputs 1, is identical. That is, we have $\Pr[\mathsf{Expt}^{\mathtt{normal}}_{\mathcal{C},\mathcal{B}}(\lambda) = 1 | u = j + 1] = \Pr[\mathsf{Expt}^{\mathtt{eq}}_{\mathcal{C},\mathcal{B}}(\lambda) = 1 | u = j]$.

Hence, we can compute $\mathcal{B}$'s advantage against the indistinguishability of the two modes as follows.

$$\begin{aligned}
\mathsf{Adv}^{\mathtt{ind}}_{\mathcal{C},\mathcal{B}}(\lambda) &= \Big|\Pr[\mathsf{Expt}^{\mathtt{normal}}_{\mathcal{C},\mathcal{B}}(\lambda) = 1] - \Pr[\mathsf{Expt}^{\mathtt{eq}}_{\mathcal{C},\mathcal{B}}(\lambda) = 1]\Big| \\
&= \Big|\sum_{j \in [\kappa]} \Pr[u = j] \cdot \Big(\Pr[\mathsf{Expt}^{\mathtt{normal}}_{\mathcal{C},\mathcal{B}}(\lambda) = 1 | u = j] - \Pr[\mathsf{Expt}^{\mathtt{eq}}_{\mathcal{C},\mathcal{B}}(\lambda) = 1 | u = j]\Big)\Big| \\
&= \frac{1}{\kappa} \cdot \Big|\Pr[\mathsf{Expt}^{\mathtt{normal}}_{\mathcal{C},\mathcal{B}}(\lambda) = 1 | u = 1] - \Pr[\mathsf{Expt}^{\mathtt{eq}}_{\mathcal{C},\mathcal{B}}(\lambda) = 1 | u = \kappa]\Big| \\
&= \Big|\Pr[\mathsf{S}_1] - \Pr[\mathsf{S}_2]\Big|.
\end{aligned}$$

Since $\mathcal{C}$ satisfies the indistinguishability of the two modes, $\mathsf{Adv}^{\mathtt{ind}}_{\mathcal{C},\mathcal{B}}(\lambda)$ is negligible, and thus $|\Pr[\mathsf{S}_1] - \Pr[\mathsf{S}_2]|$ must be negligible.

B.2 Game 2 vs. Game 3

Here, we show that $|\Pr[\mathsf{S}_2] - \Pr[\mathsf{S}_3]|$ is negligible due to the CPA security of the underlying PKE scheme Π.

Let $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ be the wCCA adversary considered in the proof of Theorem 5. Using $\mathcal{A}$ as a building block, we construct a reduction algorithm $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ attacking the CPA security of the underlying PKE scheme Π.

$\mathcal{B}_1(pk')$**:** $\mathcal{B}_1$ initially runs $(\mathsf{t}^*, \mathsf{st}_0) \leftarrow \mathcal{A}_0(1^\lambda)$. Then, $\mathcal{B}_1$ picks $\mathbf{s}^* = (s^*_1, \ldots, s^*_\kappa) \in \{0,1\}^\kappa$ and $u \in [\kappa]$ uniformly at random. For each $i \in [\kappa]$, $\mathcal{B}_1$ computes $(\widetilde{ck}_i, \widetilde{\gamma}_i, \widetilde{\rho}^0_i, \widetilde{\rho}^1_i) \leftarrow \mathsf{EQSetup}(1^\lambda, \mathsf{t}^*)$, and sets $(ck_i, \gamma^*_i, \rho^{*0}_i, \rho^{*1}) \leftarrow (\widetilde{ck}_i, \widetilde{\gamma}_i, \widetilde{\rho}^0_i, \widetilde{\rho}^1)$. Then, $\mathcal{B}_1$ sets $m'_0 = \bot$, $m'_1 = \rho^{*1}_u$, Finally, $\mathcal{B}_1$ sets $\mathsf{st}_\mathcal{B}$ as all the values known to $\mathcal{B}_1$, and terminates with output $(m'_0, m'_1, \mathsf{st}_\mathcal{B})$.

$\mathcal{B}_2(c', \mathsf{st}_\mathcal{B})$**:** $\mathcal{B}_2$ generates the remaining key materials and the components of the challenge ciphertexts C^* (except for c^*_m) as follows:

1. Compute $(pk^0, sk^0) \leftarrow \mathsf{KG}(1^\lambda)$, and set $pk^1 \leftarrow pk'$.
2. Compute $crs \leftarrow \mathsf{CRSG}(1^\lambda, \kappa, n)$.
3. Pick $H \leftarrow \mathcal{H}$.
4. Set $PK = (pk^0, pk^1, (ck_i)_i, crs, H)$ and $SK = (sk^0, PK)$.
5. For each $i \in [\kappa]$, do the following:
   (a) Pick $r^{*0}_i, r^{*1}_i \leftarrow \mathcal{R}_\lambda$.
   (b) Generate (c^{*0}_i, c^{*1}_i) as follows, based on the relation between i and u:
   - Case $i < u$:
     i. $(M^{*0}_i,\ M^{*1}_i) \leftarrow (\rho^{*0}_i, \rho^{*1}_i)$
     ii. For both $v \in \{0,1\}$: $c^{*v}_i \leftarrow \mathsf{Enc}(pk^v, M^{*v}_i; r^{*v}_i)$.
   - Case $i > u$:
     i. $(M^{*0}_i,\ M^{*1}_i) \leftarrow \begin{cases} (\rho^{*0}_i, \bot) & \text{if } s^*_i = 0 \\ (\bot, \rho^{*1}_i) & \text{if } s^*_i = 1 \end{cases}$
     ii. For both $v \in \{0,1\}$: $c^{*v}_i \leftarrow \mathsf{Enc}(pk^v, M^{*v}_i; r^{*v}_i)$.
   - Case $i = u$:
     i. $c^{*0}_u \leftarrow \begin{cases} \mathsf{Enc}(pk^0, \rho^{*0}_u; r^{*0}_u) & \text{if } s^*_u = 0 \\ \mathsf{Enc}(pk^0, \bot; r^{*0}_u) & \text{if } s^*_u = 1 \end{cases}$.
     ii. $c^{*1}_u \leftarrow \begin{cases} c' \ \ (\mathcal{B}\text{'s challenge ciphertext}) & \text{if } s^*_u = 0 \\ \mathsf{Enc}(pk^1, \rho^{*1}_u; r^{*1}_u) & \text{if } s^*_u = 1 \end{cases}$.
     Note that $\mathcal{B}$ embeds its challenge ciphertext c' into c^{*1}_u only when $s^*_u = 0$.
   (c) Set $x^*_i = (pk^0, pk^1, ck_i, \mathsf{t}^*_i, c^{*0}_i, c^{*1}_i, \gamma^*_i)$ and $w^*_i = (s^*_i, \rho^{*(s^*_i)}_i, r^{*(s^*_i)}_i)$.[14]
6. Compute $\pi^* \leftarrow \mathsf{BProve}(crs, (x^*_i)_i, (w^*_i)_i)$.

[14] Note that this step does not require r^{*1}_u in case $s^*_u = 0$.

Then, $\mathcal{B}_2$ runs $(m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}_1(PK, \mathsf{st}_0)$, where $\mathcal{B}_2$ answers $\mathcal{A}_1$'s decryption queries using SK. $\mathcal{B}_2$ picks the challenge bit $b \leftarrow \{0,1\}$, computes $c^*_m \leftarrow H(\mathbf{s}^*) \oplus m_b$, and sets $C^* = ((c_i^{*0}, c_i^{*1}, \gamma_i^*)_i, \pi^*, c^*_m)$. $\mathcal{B}_2$ then runs $b' \leftarrow \mathcal{A}_2(C^*, \mathsf{st})$, where $\mathcal{A}_2$'s decryption queries are responded as before. Finally, at some point, $\mathcal{A}_2$ terminates with its guess bit b' for b. $\mathcal{B}_2$ sets $\beta' \leftarrow (b' \stackrel{?}{=} b)$, and terminates with output β'.

The above completes the description of $\mathcal{B}$. Let $\beta \in \{0,1\}$ denote the challenge bit in $\mathcal{B}$'s CPA security experiment. It is not hard to see the following facts:

- If $\beta = 0$ and the value u picked by $\mathcal{B}_1$ at the beginning is $u = 1$, then c_1^{*0} $\mathcal{B}$ simulates Game 2 perfectly for $\mathcal{A}$. Therefore, the probability that $\mathcal{B}$ outputs $\beta' = 1$ is equivalent to the probability that $\mathcal{A}$ succeeds in Game 2, namely, $\Pr[\beta' = 1 | \beta = 0 \wedge u = 1] = \Pr[\mathsf{S}_2]$.
- If $\beta = 1$ and the value u picked by $\mathcal{B}_1$ at the begining is $u = \kappa$, then $\mathcal{B}$ simulates Game 3 perfectly for $\mathcal{A}$. Therefore, the probability that $\mathcal{B}$ outputs $\beta' = 1$ is equivalent to the probability that $\mathcal{A}$ succeeds in Game 2, namely, $\Pr[\beta' = 1 | \beta = 1 \wedge u = \kappa] = \Pr[\mathsf{S}_3]$.
- From $\mathcal{A}$'s view point, for any $j \in \{1, \ldots, \kappa - 1\}$, the experiment simulated by $\mathcal{B}$ in the following situations is identically distributed:
  - $\beta = 1$ and the value u picked by $\mathcal{B}_1$ at the beginning is $u = j + 1$.
  - $\beta = 0$ and the value u picked by $\mathcal{B}_1$ at the beginning is $u = j$.

  Hence, in these situations, the probability that $\mathcal{A}$ succeeds in guessing the challenge bit, and consequently $\mathcal{B}$ outputs 1, is identical. That is, we have $\Pr[\beta' = 1 | \beta = 0 \wedge u = j + 1] = \Pr[\beta' = 1 | \beta = 1 \wedge u = j]$.

Hence, we can compute $\mathcal{B}$'s advantage against the CPA security is as follows.

$$\begin{aligned}
\mathsf{Adv}^{\mathtt{CPA}}_{\Pi,\mathcal{B}}(\lambda) &= \Big|\Pr[\beta' = 1 | \beta = 0] - \Pr[\beta' = 1 | \beta = 1]\Big| \\
&= \Big|\sum_{j \in [\kappa]} \Pr[u = j] \cdot \Big(\Pr[\beta' = 1 | \beta = 0 \wedge u = j] - \Pr[\beta' = 1 | \beta = 1 \wedge u = j]\Big)\Big| \\
&= \frac{1}{\kappa} \cdot \Big|\Pr[\beta' = 1 | \beta = 0 \wedge u = 1] - \Pr[\beta' = 1 | \beta = 1 \wedge u = \kappa]\Big| \\
&= \Big|\Pr[\mathsf{S}_2] - \Pr[\mathsf{S}_3]\Big|.
\end{aligned}$$

Since Π is CPA secure, $\mathsf{Adv}^{\mathtt{CPA}}_{\Pi,\mathcal{B}}(\lambda)$ is negligible, and thus $|\Pr[\mathsf{S}_2] - \Pr[\mathsf{S}_3]|$ must be negligible.

C Definitions for Reusable Privately-Verifiable BARG

Here, we give the formal definitions for a *privately-verifiable* BARG and its *reusable* semi-adaptive soundness.

Let $\mathcal{L} \subseteq \{0,1\}^*$ be an NP language. A privately-verifiable non-interactive batch argument system in the common reference string (CRS) model (referred to simply as *privately-verifiable BARG*) for $\mathcal{L}$ consists of the three PPT algorithms $(\mathsf{CRSG}, \mathsf{BProve}, \mathsf{BVerify})$ with the following syntax:

Setup:	$(crs, vk) \leftarrow \mathsf{CRSG}(1^\lambda, k, n)$
Proof Generation:	$\pi \leftarrow \mathsf{BProve}(crs, (x_i)_{i\in[k]}, (w_i)_{i\in[k]})$
Verification:	$\top / \bot \leftarrow \mathsf{BVerify}(vk, (x_i)_{i\in[k]}, \pi)$

where vk is a verification key (corresponding to the CRS crs), and the rest of the values are the same as in for a (publicly-verifiable) BARG in Section 3.1.

The correctness (completeness) and ℓ-succinctness of a privately-verifiable BARG are defined analogously to those for a (publicly-verifiable) BARG in Section 3.1, and thus omitted.

Reusable Semi-adaptive Soundness. Let $\mathcal{P} = (\mathsf{CRSG}, \mathsf{BProve}, \mathsf{BVerify})$ be a BARG for an NP language $\mathcal{L}$. For $\mathcal{P}$ and an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, we define the reusable semi-adaptive soundness experiment $\mathsf{Expt}^{\texttt{r-sa-sound}}_{\mathcal{P},\mathcal{A}}(\lambda)$ as follows.

$\mathsf{Expt}^{\texttt{r-sa-sound}}_{\mathcal{P},\mathcal{A}}(\lambda)$:
$(k, n, i^*, \mathsf{st}) \leftarrow \mathcal{A}_1(1^\lambda)$ // k and n are polynomials, $i^* \in [k(\lambda)]$
$(crs, vk) \leftarrow \mathsf{CRSG}(1^\lambda, k, n)$
$((x_i)_{i\in[k(\lambda)]}, \pi) \leftarrow \mathcal{A}_2^{\mathsf{BVerify}(vk,\cdot,\cdot)}(crs, \mathsf{st})$
If **(a)** $\wedge$ **(b)** $\wedge$ **(c)** then return 1 else return 0:
(a) $\mathsf{BVerify}(vk, (x_i)_{i\in[k(\lambda)]}, \pi) = \top$
(b) $x_{i^*} \notin \mathcal{L}$
(c) $\forall i \in [k(\lambda)] : |x_i| \leq n(\lambda)$

Definition 7. *We say that a privately-verifiable BARG $\mathcal{P}$ satisfies reusable semi-adaptive soundness if for all PPT adversaries $\mathcal{A}$,* $\mathsf{Adv}^{\texttt{r-sa-sound}}_{\mathcal{P},\mathcal{A}}(\lambda) := \Pr[\mathsf{Expt}^{\texttt{r-sa-sound}}_{\mathcal{P},\mathcal{A}}(\lambda) = 1]$ *is negligible.*

Table of Contents