Revisiting Security Definitions of Sender-Anamorphic Encryption

Yuichi Tanishita[1,2], Takahiro Matsuda[2], and Kanta Matsuura[1]

[1] The University of Tokyo, Tokyo, Japan
y-tani@iis.u-tokyo.ac.jp, kanta@iis.u-tokyo.ac.jp,
[2] National Institute of Advanced Industrial Science and Technology, Tokyo, Japan
t-matsuda@aist.go.jp

Abstract. Sender-anamorphic encryption is a cryptographic primitive that allows a sender to covertly embed an alternative message into the ciphertext. This enables the sender to transmit the message they truly wish to send without an authority's knowledge, even if they are coerced into sending a message against their will. The concrete scenario considered here is one where the authority demands that the sender provide the public key, the plaintext, and the internal randomness used to generate the ciphertext, and then requires a proof that the coerced message was indeed encrypted correctly.

Persiano et al. (Eurocrypt 2022) formulated the security of sender-anamorphic encryption to capture this situation. Building on that, Wang et al. (Asiacrypt 2023) proposed ℓ-sender-anamorphic encryption along with its security definition. However, in the formal security definitions for sender-anamorphic encryption in these existing works, the randomness used to generate the challenge ciphertext is not given to an adversary, and thus, the potential threats are not fully accounted for.

Therefore, in this study, we redefine security for sender-anamorphic encryption so that the randomness used to generate the challenge ciphertext is provided to the adversary. We then investigate whether the existing sender-anamorphic encryption schemes by Persiano et al. and Wang et al. satisfy our refined notions of security.

Keywords: Anamorphic encryption · Public-key encryption · Randomness disclosure

Table of Contents

1 Introduction . . . 3
1.1 Our Contributions . . . 4
1.2 Related Works . . . 6
2 Preliminaries . . . 8
2.1 Properties that Generally Hold for Probability Distributions . . . 9
2.2 Public-Key Encryption . . . 9
2.3 Key Encapsulation Mechanism . . . 10
2.4 Data Encapsulation Mechanism . . . 11
2.5 Pseudorandom Function . . . 12
2.6 Entropy-Smoothing Function . . . 12
3 Refining Security Definitions of Sender-Anamorphic Encryption . . . 13
3.1 Syntax and Correctness . . . 13
3.2 Existing Security Definitions . . . 14
3.3 Our Refined Security Definitions . . . 14
4 Persiano et al.'s SAME Scheme . . . 16
5 Wang et al.'s First SAME Scheme and Its Modified Scheme . . . 18
5.1 Wang et al.'s First Scheme and Its SAME-R Security . . . 18
5.2 Modifying Wang et al.'s First Scheme for S-SAME-R Security . . . 27
6 Wang et al.'s Second SAME Scheme . . . 39
A Proofs of Lemmas in Section 2 . . . 54
A.1 Proof of Lemma 1 . . . 54
A.2 Proof of Lemma 2 . . . 54
B Proof of Theorem 2 in Section 3 . . . 55
C Robustness . . . 56
C.1 Robustness of Each Scheme . . . 57
D Proofs of Theorem 6 and Lemmas in Section 5 . . . 58
D.1 Proof of Theorem 6 . . . 58
D.2 Proof of Lemma 3 . . . 59
D.3 Proof of Lemma 4 . . . 63
D.4 Proof of Lemma 6 . . . 64
D.5 Proof of Lemma 7 . . . 66
D.6 Proof of Lemma 11 . . . 68
D.7 Proof of Lemma 12 . . . 71
D.8 Proof of Lemma 13 . . . 73
D.9 Proof of Lemma 14 . . . 75
E Special PRF Property of Hashed ElGamal KEM . . . 76

1 Introduction

While encryption is a powerful tool for protecting message privacy, its security implicitly relies on two assumptions: *Receiver Privacy* that a receiver keeps the secret key confidential, and *Sender Freedom* that a sender can freely choose any message and any public key to encrypt. Persiano et al. [26] pointed out that both assumptions can be compromised by a dictator. For example, receiver privacy can be compromised if a government forces citizens to disclose their secret keys, and sender freedom can be compromised if an authority forces users to encrypt a designated message or implicitly restricts them from sending any messages they desire.

To address such scenarios, Persiano et al. introduced the concept of *anamorphic encryption* (AME), which enables secure communication even when either assumption is compromised. The idea of AME is embedding a hidden message (called a *duplicate message* in this context) into an ordinary ciphertext, while making the ciphertext appear legitimate to a dictator. Specifically, there are two types of anamorphic encryption: *receiver-anamorphic encryption* (RAME), which is secure against compromise of receiver privacy, and *sender-anamorphic encryption* (SAME), which is secure against compromise of sender freedom. In this work, we focus on the latter.

In the threat model underlying SAME, a dictator forces a sender to send a designated message fm (that stands for a *forced message*) to a designated receiver with its public key fpk (that stands for a *forced public key*), so the sender cannot freely choose either of the message or the receiver. Even in such a situation, SAME allows the sender to secretly embed a duplicate message dm into the ciphertext that the sender is forced to compute by the dictator by using a fake randomness generated from $\mathsf{fpk}, \mathsf{fm}, \mathsf{dm}$, and a duplicate public key dpk, where $(\mathsf{dpk}, \mathsf{dsk})$ is the key pair of another (legitimate) receiver to whom the sender is actually trying to communicate dm. Such a ciphertext is called an *anamorphic ciphertext*. To the dictator, the anamorphic ciphertext looks like an encryption of fm under fpk, which can be decrypted to fm by the designated receiver. However, the same ciphertext can also be decrypted to the hidden duplicate message dm by the legitimate receiver with dsk.

In the original model of SAME [26], a single ciphertext of the forced message fm conveys a single duplicate message dm. Wang et al. [29] extended (relaxed) this setting by introducing *ℓ-sender anamorphic encryption* (ℓ-SAME), where a single duplicate message dm is transmitted across ℓ ciphertexts $\mathsf{c}_1, \ldots, \mathsf{c}_\ell$ that respectively correspond to ℓ forced messages $\mathsf{fm}_1, \ldots, \mathsf{fm}_\ell$. (Hence, SAME originally considered by Persiano et al. is 1-SAME in the definition of Wang et al.. Looking ahead, in this work, we will mainly treat this ℓ-sender variant.) Conceptually, an anamorphic encryption scheme extends a public-key encryption (PKE) scheme by incorporating a fake-randomness generation algorithm and an algorithm that recovers dm from anamorphic ciphertexts.

Technically speaking, in the threat model of SAME, the sender is required to produce a ciphertext using the dictator's designated fpk and fm, and to submit both the resulting ciphertext and the randomness used for encryption to prove

that he has followed the order of the dictator. As long as the dictator cannot detect that the ciphertext was computed irregularly, the sender can secretly communicate dm even under the dictator's surveillance. In fact, Persiano et al. [26] describe this situation as "*the more remote adversary does select the message to be sent, lets Alice compute the ciphertext ct, and later on Alice is required to exhibit the coin tosses used to compute ct*". This clearly indicates that Persiano et al. consider the situation in which the adversary (dictator) obtains the randomness used for encryption (of forced messages).

However, in the actual formal security definition for SAME given by Persiano et al. [26, Section 6], the randomness used in encryption is *not* given to the adversary. More specifically, in their formal security definition for SAME, the adversary receives fpk, and can access the encryption oracle that, upon receiving a pair of messages $(\mathsf{fm}, \mathsf{dm})$, returns a ciphertext of fm generated as follows, depending on the challenge bit:

- an ordinary (i.e. non-anamorphic) ciphertext of fm generated using an honestly sampled randomness, or
- an anamorphic ciphertext of fm generated using a fake randomness computed from the public parameters, fpk, fm, dpk, and dm. (This ciphertext can be decrypted to fm using fsk, while it can also be decrypted to dm using dsk.)

Security for SAME is then defined as the indistinguishability between these two cases. However, the oracle's response includes only the ciphertexts, not the corresponding randomness, which makes it impossible for the adversary to perform the intended verification. This is inconsistent with what they describe in their texts. Consequently, the dictator is not guaranteed to be able to verify whether the ciphertext was generated using the genuine randomness corresponding to $(\mathsf{fpk}, \mathsf{fm})$, and thus their formal definition fails to capture the intended threat model. The formal security definition for ℓ-SAME given by Wang et al. [29] inherits the same issue. Therefore, the existing security definitions for (ℓ-)SAME do not fully capture the threats that SAME was originally designed to address.

In this work, we revisit the formalization of SAME to provide a security definition that accurately captures the threat of compromised sender freedom.

1.1 Our Contributions

In this work, we introduce refined security notions for (ℓ-)SAME that accurately capture the threats it is originally intended to address, and then investigate whether the existing constructions satisfy our new definitions.

Refining Security Definitions. Our contributions regarding the formalization of security are as follows, which are given in Section 3.

1. We define the new security notion for (ℓ-)SAME that we call `SAME-R` security. In the `SAME-R` security experiment, the adversary obtains not only the ciphertexts but also the randomness used in encryption. Furthermore, we also define the stronger version of the security notion that we call `S-SAME-R`

security. S-SAME-R security is identical to SAME-R security except that the adversary additionally learns fsk and dpk, in addition to fpk.
2. We introduce variants of our new notions, which we call SAME-R′ security and S-SAME-R′ security. In the security experiments, the adversary obtains only the randomness used in encryption (but not the ciphertext itself). S-SAME-R′ is a stronger version of SAME-R′. (The difference between SAME-R′ and S-SAME-R′ is analogous to the one between SAME-R and S-SAME-R.)
3. We show that SAME-R (resp. S-SAME-R) and SAME-R′ (resp. S-SAME-R′) are equivalent.

Our new security notions are based on the security definitions for ℓ-SAME proposed by Wang et al. [29] (which is in turn based on the one for SAME by Persiano et al. [26]). In Wang et al.'s definition for ℓ-SAME, a duplicate message dm is transmitted across ℓ ciphertexts $\mathsf{c}_1, \ldots, \mathsf{c}_\ell$ corresponding to forced messages $\mathsf{fm}_1, \ldots, \mathsf{fm}_\ell$. Their security notion captures the indistinguishability between ordinary PKE ciphertexts and anamorphic ciphertexts. Specifically, in their game, the adversary, given public parameters and the forced public key fpk, can access the encryption oracle with any pair of fm and dm. The oracle, depending on the challenge bit, either generates true randomness or fake randomness, uses it to encrypt fm, and returns the resulting ciphertext. The adversary finally outputs a guess bit, and the scheme is secure if its advantage is negligible. In our definition, the oracle's response includes not only the ciphertext but also the randomness used in its generation. This modification faithfully represents the scenario where the sender is under the surveillance of an authority that can demand both the ciphertext and the randomness, yet cannot detect whether a duplicate message is embedded. Thus, the new definition accurately models the threat.

We also introduce SAME-R′ and S-SAME-R′ security, a variant of SAME-R and S-SAME-R, respectively, in which the oracle's response consists only of the randomness, excluding the ciphertext. Although SAME-R′ and S-SAME-R′ security do not capture any new threat beyond SAME-R and S-SAME-R, it is often useful for concise proofs. Since we can prove that SAME-R (resp. S-SAME-R) is equivalent to SAME-R′ (resp. S-SAME-R′), it is sufficient to show security analysis in SAME-R′ or S-SAME-R′ in which the components of oracle responses are fewer.

For the details on our refined security notions for SAME, see Section 3.

Existing Constructions under New Security Notions. We investigate whether the existing SAME constructions satisfy our refined security notions. The schemes we analyzed are:

- the 1-sender anamorphic encryption scheme by Persiano et al. [26];
- the ℓ-sender PKE-based anamorphic encryption scheme by Wang et al. [29]; and
- the $(\ell + 1)$-sender hybrid-encryption-based scheme by Wang et al [29].

Our findings are summarized as follows:

- In Section 4, we show that the rejection-sampling-based scheme by Persiano et al. [26, Section 6] satisfies SAME-R security. However, we show that this scheme is not S-SAME-R secure by showing an efficient attack.

– In Section 5, we show that the ℓ-sender PKE-based anamorphic encryption scheme by Wang et al. [29, Section 4] (which we also concisely call Wang et al.'s first scheme), satisfies SAME-R security under the same assumption as used for proving the original security notion.
– In Section 6, we show that the $(\ell+1)$-sender hybrid-encryption-based scheme by Wang et al. [29, Section 5] (which we also concisely call Wang et al.'s second scheme) satisfies SAME-R security, if we add some mild assumption on one of its building blocks. However, we show that this scheme is not S-SAME-R secure by showing an efficient attack.
– For Wang et al.'s first scheme, we investigate whether it satisfies S-SAME-R security. Although the original proof technique makes it difficult to directly establish this, we show that by slightly modifying the scheme, replacing the underlying PKE with hybrid encryption constructed from a key encapsulation mechanism (KEM) and a data encapsulation mechanism (DEM) with pseudorandom ciphertexts, and assuming that the hash function is a pseudorandom function (PRF), the resulting scheme achieves S-SAME-R security. (The rationale behind these modifications is discussed in Section 5.2.)

Through these analyses, we demonstrate that the existing SAME schemes indeed satisfy the appropriate form of security that properly reflects the intended threat model. Moreover, our work realizes that while some schemes achieve the strong version of the security, others do not, clarifying the boundary between standard and strong SAME security.

1.2 Related Works

Sender-Anamorphic Encryption. The concept of SAME was first introduced by Persiano et al. [26], who formalized its security and correctness and proposed a one-bit SAME scheme based on rejection sampling. Wang et al. [29] later relaxed the original model, in which a single duplicate message dm is embedded into a single ciphertext of a forced message fm, and extended it to the ℓ-sender setting, where one duplicate message is embedded across ℓ ciphertexts of forced messages $\mathsf{fm}_1, \ldots, \mathsf{fm}_\ell$. They further formalized the notions of security and robustness under this extended model. Here, robustness means that the anamorphic decryption algorithm outputs a special symbol $\perp$ when given (i) a regular ciphertext with a duplicate key, or (ii) an anamorphic ciphertext with a regular secret key together. This property ensures that the anamorphic decryption algorithm produces a valid output only when it is provided with a matching anamorphic ciphertext and the corresponding secret key.

Wang et al. presented two constructions that achieve both security and robustness and established theoretical connections between SAME and other related primitives, including RAME, public stegosystems, and asymmetric algorithm-substitution attacks.[3]

[3] For more details of the relationships and related primitives, see [29].

Receiver-Anamorphic Encryption. The threat scenario considered in RAME concerns a compromise of receiver privacy. Specifically, the receiver's key pair $(\mathsf{apk}, \mathsf{ask})$ used for public-key communication is seized by a dictator, and all ciphertexts sent to the receiver are subject to censorship. Even in such a setting, if the sender and the receiver secretly share a duplicate key dk unknown to the dictator, secure communication can still be achieved. The sender encrypts the forced message fm together with the duplicate message dm, the public key apk, and the duplicate key dk to produce a ciphertext. Although the dictator, using ask, can decrypt only fm and cannot detect the presence of any covert message, the legitimate receiver can use dk to decrypt the same ciphertext and recover dm. Thus, communication remains private even when receiver privacy is compromised.

The concept of RAME was first introduced by Persiano et al. [26], who formalized its syntax and security definitions. In their model, the duplicate key dk is generated together with the receiver's key pair $(\mathsf{apk}, \mathsf{ask})$. Consequently, once a dictator gains power over an already-established communication channel, it is no longer possible to retroactively create an anamorphic channel on top of it. To address this limitation, Banfi et al. [5] proposed an extended model in which dk can be generated after regular key generation. They also introduced a notion of robustness, which requires that both the regular and the anamorphic decryption algorithms output a special symbol $\perp$ when invoked with an incorrect pair of ciphertext and key, namely regular ciphertext and duplicate key, or anamorphic ciphertext and regular key. Robustness ensures that decryption succeeds only with a matching key–ciphertext pair. Several concrete constructions satisfying these properties have since been proposed [5, 10, 25, 26].

In RAME, the sender and receiver must share a duplicate key without the dictator's recognition. This may be difficult in practice, depending on the communication setting, motivating additional research on key-exchange protocols that support such key sharing [28].

Most existing RAME schemes conceptually overlay a symmetric-key–based anamorphic channel on top of a PKE-based regular channel. To enable fine partitioning between the encryption and decryption keys, asymmetric AME was proposed by Catalano et al. [8], where distinct keys are used for the two channels. Although both asymmetric RAME and SAME appear similar in the sense that they both are PKE-based syntax for an anamorphic channel, they capture fundamentally different threats: asymmetric RAME addresses compromise of receiver privacy, whereas SAME focuses on sender freedom. Recent works of asymmetric RAME have produced a variety of results on the security and constructions [4, 9, 10, 15, 27]. A comprehensive study analyzing both symmetric and asymmetric RAME, along with their theoretical relationships, was presented in [11]. Moreover, quantum RAME has also been explored [17].

Finally, because the goal of anamorphic encryption is to conceal the existence of a covert channel from the dictator, its constructions must be built upon well-known and widely used encryption schemes. Therefore, examining some properties or constructability of RAME is important. In contrast, Dodis et al. [16]

introduced the notion of anamorphic-resistant encryption (ARE), which characterizes encryption schemes that cannot be made anamorphic, and subsequent works have investigated ARE under various settings [3, 6].

Other Anamorphic Primitives. The concept of anamorphic encryption has been extended to other cryptographic primitives as well. A representative example is anamorphic signatures [24], which enable a covert channel even when all signed messages must be transmitted under the surveillance of a dictator. In such settings, signatures are typically used to prove message integrity and authenticity; however, by embedding a duplicate message into the signature, the signer can establish a covert channel that remains undetectable to the dictator. Several extensions and concrete constructions of anamorphic signatures have been explored [4, 12, 14, 21, 24].

2 Preliminaries

In this section, we review the basic notation and the formal definitions for existing cryptographic primitives and their security properties used in this paper.

Basic Notation. $\mathbb{N}$ denotes the set of all natural numbers. Throughout the paper, we will use $\lambda \in \mathbb{N}$ to denote the security parameter, which will sometimes be written in its unary representation, 1^λ. Furthermore, we sometimes suppress the dependency on λ, when λ is clear from the context. If $n \in \mathbb{N}$, then we define $[n] := \{1, \ldots, n\}$. Let $\mathsf{FUNC}_{\mathcal{X}\to\mathcal{Y}}$ be the set of all functions whose domain is $\mathcal{X}$ and range is $\mathcal{Y}$. For strings x and y, $|x|$ denotes the bit length of x, $x||y$ denotes the concatenation of x and y, and $(x \stackrel{?}{=} y)$ denotes the operation that returns 1 if and only if $x = y$. For a set S, $x \xleftarrow{\$} S$ denotes that an element x is chosen uniformly at random from S. $\{\mathrm{op1}, \mathrm{op2}, \cdots : x\}$ denotes the distribution of x after performing the sequence of operations op1, op2, ... For distributions $\mathcal{D}_0$ and $\mathcal{D}_1$, $\mathcal{D}_0 \equiv \mathcal{D}_1$ denotes that $\mathcal{D}_0$ is equivalent to $\mathcal{D}_1$. $y \leftarrow x$ denotes that x is deterministically assigned to the variable y. If M is a probabilistic algorithm, then $y \xleftarrow{\$} \mathsf{M}(x)$ denotes that M receives the input x and outputs y. When we need to make the randomness r used by M explicit, we denote $y \leftarrow \mathsf{M}(x; r)$ (in which case, the computation by M is deterministic on input x and r). If $\mathcal{O}$ is a probabilistic algorithm or a function, then $\mathsf{M}^{\mathcal{O}}$ denotes that M has oracle access to $\mathcal{O}$. A function $f(\lambda) : \mathbb{N} \to [0, 1]$ is called negligible if for any constants $c \in \mathbb{N}$ and all sufficiently large $\lambda \in \mathbb{N}$, we have $f(\lambda) < 1/\lambda^c$. PPT stands for probabilistic polynomial time.

Statistical Distance. Let $\mathcal{X}$ and $\mathcal{Y}$ be distributions defined over a finite set S. The statistical distance between $\mathcal{X}$ and $\mathcal{Y}$ is defined as $\mathbf{SD}(\mathcal{X}, \mathcal{Y}) := \frac{1}{2}\sum_{z\in S} |\Pr[\mathcal{X} = z] - \Pr[\mathcal{Y} = z]|$.

2.1 Properties that Generally Hold for Probability Distributions

In this paper, we will use the following simple facts that generally hold for probability distributions. For completeness, formal proofs for them are given in Appendix A.

Lemma 1. *Let X be a finite set, and let $\mathsf{P} : X \to \{0,1\}$ be an arbitrary predicate such that $\mathsf{P}^{-1}(1) \neq \emptyset$. Then,*

$$\left\{h \leftarrow 0;\ \textit{do}\ \{h \leftarrow h+1;\ x_h \xleftarrow{\$} X\}\ \textit{until}\ \mathsf{P}(x_h) = 1 : x_h\right\} \equiv \left\{x \xleftarrow{\$} \mathsf{P}^{-1}(1) : x\right\}.$$

Lemma 2. *Let X and Y be finite sets and $f : X \to Y$ be a function. Then,*

$$\left\{x \xleftarrow{\$} X : x\right\} \equiv \left\{x \xleftarrow{\$} X;\ y \leftarrow f(x);\ \widehat{x} \xleftarrow{\$} f^{-1}(y) : \widehat{x}\right\}.$$

2.2 Public-Key Encryption

Here, we review the definitions for a public-key encryption (PKE) scheme.

A PKE scheme PKE consists of the following four PPT algorithms (Setup, $\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec}$).

$\mathsf{Setup}(1^\lambda) \xrightarrow{\$} \mathsf{pp}$**:** This is the setup algorithm that takes a security parameter 1^λ as input, and outputs a public parameter pp.

$\mathsf{KG}(\mathsf{pp}) \xrightarrow{\$} (\mathsf{pk}, \mathsf{sk})$**:** This is the key generation algorithm that takes a public parameter pp as input, and outputs a pair of public key pk and secret key sk.

$\mathsf{Enc}(\mathsf{pk}, \mathsf{m}) \xrightarrow{\$} \mathsf{c}$**:** This is the encryption algorithm that takes a public key pk and a plaintext m as input, and outputs a ciphertext c.

$\mathsf{Dec}(\mathsf{sk}, \mathsf{c}) \to \mathsf{m}'/\perp$**:** This is the decryption algorithm that takes a secret key sk and a ciphertext c as input, and outputs a decrypted plaintext m', or $\perp$ to indicate a decryption failure.

Correctness. We require that $\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{m})) = \mathsf{m}$ holds for any $\lambda \in \mathbb{N}$, $\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$, $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp})$, and plaintext m.

Security Notions. In this paper, we will treat the following two kinds of security notions for a PKE scheme.

Definition 1 ($\mathtt{IND\text{-}CPA}$ Security of PKE). *We say that PKE is $\mathtt{IND\text{-}CPA}$ secure, if for any PPT adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$, $\mathsf{Adv}^{\mathtt{IND\text{-}CPA}}_{\mathsf{PKE},\mathcal{A}}(\lambda) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{IND\text{-}CPA}}_{\mathsf{PKE},\mathcal{A}}(\lambda) = 1] - \frac{1}{2}|$ is negligible, where the experiment $\mathsf{Expt}^{\mathtt{IND\text{-}CPA}}_{\mathsf{PKE},\mathcal{A}}(\lambda)$ is defined as in Fig. 1 (left).*

Definition 2 (Pseudorandomness of PKE). *We say that PKE satisfies* pseudorandomness, *if for any PPT adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$, $\mathsf{Adv}^{\mathtt{PR}}_{\mathsf{PKE},\mathcal{A}}(\lambda) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{PR}}_{\mathsf{PKE},\mathcal{A}}(\lambda) = 1] - \frac{1}{2}|$ is negligible, where the experiment $\mathsf{Expt}^{\mathtt{PR}}_{\mathsf{PKE},\mathcal{A}}(\lambda)$ is defined as in Fig. 1 (right).*

$\underline{\mathsf{Expt}^{\mathtt{IND\text{-}CPA}}_{\mathsf{PKE},\mathcal{A}}(\lambda)}$:
$b \xleftarrow{\$} \{0,1\}$
$\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$
$(\mathsf{pk},\mathsf{sk}) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp})$
$(\mathsf{m}_0,\mathsf{m}_1,\mathrm{st}) \xleftarrow{\$} \mathcal{A}_0(\mathsf{pp},\mathsf{pk})$
$\mathsf{c} \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk},\mathsf{m}_b)$
$b' \xleftarrow{\$} \mathcal{A}_1(\mathsf{c},\mathrm{st})$
return $(b \stackrel{?}{=} b')$

$\underline{\mathsf{Expt}^{\mathtt{PR}}_{\mathsf{PKE},\mathcal{A}}(\lambda)}$:
$b \xleftarrow{\$} \{0,1\}$
$\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$
$(\mathsf{pk},\mathsf{sk}) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp})$
$(\mathsf{m},\mathrm{st}) \xleftarrow{\$} \mathcal{A}_0(\mathsf{pp},\mathsf{pk})$
$\mathsf{c}_0 \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk},\mathsf{m})$
$\mathsf{c}_1 \xleftarrow{\$} \{0,1\}^{|\mathsf{c}_0|}$
$b' \xleftarrow{\$} \mathcal{A}_1(\mathsf{c}_b,\mathrm{st})$
return $(b \stackrel{?}{=} b')$

Fig. 1. Security experiments for PKE: The IND-CPA experiment (left) and the experiment for pseudorandomness (right). In the IND-CPA experiment, it is required that $|\mathsf{m}_0| = |\mathsf{m}_1|$.

2.3 Key Encapsulation Mechanism

Here, we review the definitions for a key encapsulation mechanism (KEM) scheme.

A KEM scheme KEM consists of the following four algorithms ($\mathsf{KSetup}, \mathsf{KKG}, \mathsf{Encap}, \mathsf{Decap}$).

$\mathsf{KSetup}(1^\lambda) \xrightarrow{\$} \mathsf{pp}$**:** This is the setup algorithm that takes a security parameter 1^λ as input, and outputs a public parameter pp.

$\mathsf{KKG}(\mathsf{pp}) \xrightarrow{\$} (\mathsf{pk},\mathsf{sk})$**:** This is the key generation algorithm that takes a public parameter pp as input, and outputs a pair of public key pk and secret key sk.

$\mathsf{Encap}(\mathsf{pk}) \xrightarrow{\$} (\mathsf{c},\mathsf{K})$**:** This is the encapsulation algorithm that takes a public key pk as input and outputs a pair of ciphertext c and session-key K.

$\mathsf{Decap}(\mathsf{sk},\mathsf{c}) \to \mathsf{K}'/\perp$**:** This is the decapsulation algorithm that takes a secret key sk and a ciphertext c as input, and outputs a decapsulated session-key K', or $\perp$ to indicate a decryption failure.

Correctness. We require that $\mathsf{Decap}(\mathsf{sk},\mathsf{c}) = \mathsf{K}$ holds for any $\lambda \in \mathbb{N}, \mathsf{pp} \xleftarrow{\$} \mathsf{KSetup}(1^\lambda)$, $(\mathsf{pk},\mathsf{sk}) \xleftarrow{\$} \mathsf{KKG}(\mathsf{pp})$, and $(\mathsf{c},\mathsf{K}) \xleftarrow{\$} \mathsf{Encap}(\mathsf{pk})$.

Security Notions. In this paper, we will treat the following three kinds of security notions for a KEM scheme.

Definition 3 ($\mathtt{IND\text{-}CPA}$ Security of KEM). *We say that* KEM *is* $\mathtt{IND\text{-}CPA}$ *secure, if for any PPT adversary* $\mathcal{A}$, $\mathsf{Adv}^{\mathtt{IND\text{-}CPA}}_{\mathsf{KEM},\mathcal{A}}(\lambda) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{IND\text{-}CPA}}_{\mathsf{KEM},\mathcal{A}}(\lambda) = 1] - \frac{1}{2}|$ *is negligible, where the experiment* $\mathsf{Expt}^{\mathtt{IND\text{-}CPA}}_{\mathsf{KEM},\mathcal{A}}(\lambda)$ *is defined as in Fig. 2 (left).*

Definition 4 (Pseudorandomness of KEM). *We say that* KEM *satisfies* pseudorandomness*, if for any PPT adversary* $\mathcal{A}$, $\mathsf{Adv}^{\mathtt{PR}}_{\mathsf{KEM},\mathcal{A}}(\lambda) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{PR}}_{\mathsf{KEM},\mathcal{A}}(\lambda)$

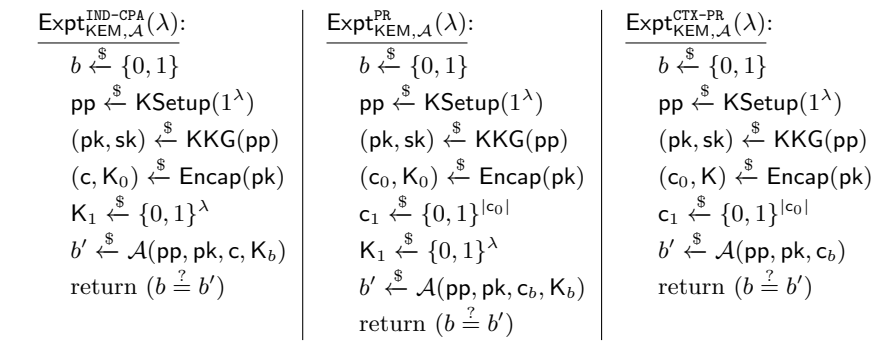

Fig. 2. Security experiments for KEM: The IND-CPA experiment (left), the experiment for pseudorandomness (center), and the experiment for ciphertext pseudorandomness (right).

$= 1] - \frac{1}{2}|$ *is negligible, where the experiment* $\mathsf{Expt}^{\mathtt{PR}}_{\mathsf{KEM},\mathcal{A}}(\lambda)$ *is defined as in Fig. 2 (center).*

Definition 5 (Ciphertext Pseudorandomness of KEM). *We say that* KEM *satisfies* ciphertext pseudorandomness*, if for any PPT adversary* $\mathcal{A}$*,* $\mathsf{Adv}^{\mathtt{CTX\text{-}PR}}_{\mathsf{KEM},\mathcal{A}}(\lambda)$ $:= 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{CTX\text{-}PR}}_{\mathsf{KEM},\mathcal{A}}(\lambda) = 1] - \frac{1}{2}|$ *is negligible, where the experiment* $\mathsf{Expt}^{\mathtt{CTX\text{-}PR}}_{\mathsf{KEM},\mathcal{A}}(\lambda)$ *is defined as in Fig. 2 (right).*

It is immediate to see that pseudorandomness implies both `IND-CPA` security and ciphertext pseudorandomness.

2.4 Data Encapsulation Mechanism

Here, we review the definitions for a data encapsulation mechanism (DEM) scheme. In this paper, we treat ones whose encryption algorithm is deterministic.

A DEM scheme DEM consists of the following two algorithms $(\mathsf{DEnc}, \mathsf{DDec})$.

$\mathsf{DEnc}(\mathsf{K},\mathsf{m}) \to \mathsf{c}$**:** This is the encryption algorithm that takes a key K and a plaintext m as input, and outputs a ciphertext c.

$\mathsf{DDec}(\mathsf{K},\mathsf{c}) \to \mathsf{m}'/\perp$**:** This is the decryption algorithm that takes a key K and a ciphertext c as input, and outputs a plaintext m', or $\perp$ to indicate a decryption failure.

Correctness. We require that $\mathsf{DDec}(\mathsf{K},\mathsf{DEnc}(\mathsf{K},\mathsf{m})) = \mathsf{m}$ holds for any key K and any plaintext m.

Security Notions. In this paper, we will treat the following two kinds of security notions for a DEM scheme.

$\underline{\mathsf{Expt}^{\mathtt{IND\text{-}OT}}_{\mathsf{DEM},\mathcal{A}}(\lambda)}$:
$b \xleftarrow{\$} \{0,1\}$
$\mathsf{K} \xleftarrow{\$} \{0,1\}^\lambda$
$(\mathsf{m}_0, \mathsf{m}_1, \mathrm{st}) \xleftarrow{\$} \mathcal{A}_0(1^\lambda)$
$\mathsf{c} \xleftarrow{\$} \mathsf{DEnc}(\mathsf{K}, \mathsf{m}_b)$
$b' \xleftarrow{\$} \mathcal{A}_1(\mathsf{c}, \mathrm{st})$
return $(b \stackrel{?}{=} b')$

$\underline{\mathsf{Expt}^{\mathtt{PR}}_{\mathsf{DEM},\mathcal{A}}(\lambda)}$:
$b \xleftarrow{\$} \{0,1\}$
$\mathsf{K} \xleftarrow{\$} \{0,1\}^\lambda$
$(\mathsf{m}, \mathrm{st}) \xleftarrow{\$} \mathcal{A}_0(1^\lambda)$
$\mathsf{c}_0 \leftarrow \mathsf{DEnc}(\mathsf{K}, \mathsf{m})$
$\mathsf{c}_1 \xleftarrow{\$} \{0,1\}^{|\mathsf{c}_0|}$
$b' \xleftarrow{\$} \mathcal{A}_1(\mathsf{c}_b, \mathrm{st})$
return $(b \stackrel{?}{=} b')$

Fig. 3. Security experiments for DEM: The IND-OT experiment (left), and the experiment for pseudorandomness (right). In the IND-OT experiment, it is required that $|\mathsf{m}_0| = |\mathsf{m}_1|$.

Definition 6 (IND-OT Security of DEM). *We say that* DEM *is* IND-OT *secure, if for any PPT adversary* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$, $\mathsf{Adv}^{\mathtt{IND\text{-}OT}}_{\mathsf{DEM},\mathcal{A}}(\lambda) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{IND\text{-}OT}}_{\mathsf{DEM},\mathcal{A}}(\lambda) = 1] - \frac{1}{2}|$ *is negligible, where the experiment* $\mathsf{Expt}^{\mathtt{IND\text{-}OT}}_{\mathsf{DEM},\mathcal{A}}(\lambda)$ *is defined as in Fig. 3 (left).*

Definition 7 (Pseudorandomness of DEM). *We say that* DEM *satisfies* pseudorandomness*, if for any PPT adversary* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$, $\mathsf{Adv}^{\mathtt{PR}}_{\mathsf{DEM},\mathcal{A}}(\lambda) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{PR}}_{\mathsf{DEM},\mathcal{A}}(\lambda) = 1] - \frac{1}{2}|$ *is negligible, where the experiment* $\mathsf{Expt}^{\mathtt{PR}}_{\mathsf{DEM},\mathcal{A}}(\lambda)$ *is defined as in Fig. 3 (right).*

2.5 Pseudorandom Function

Here, we review the definition of a pseudorandom function (PRF).

Definition 8 (Pseudorandom Function). *Let* $\mathsf{H} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ *be an efficiently computable function. We say that* H *is a* pseudorandom function*, if for any PPT adversary* $\mathcal{A}$*,*

$$\mathsf{Adv}^{\mathtt{PRF}}_{\mathsf{H},\mathcal{A}}(\lambda) := \left| \Pr_{k \xleftarrow{\$} \mathcal{K}} [\mathcal{A}^{\mathsf{H}(k,\cdot)}(1^\lambda) = 1] - \Pr_{f \xleftarrow{\$} \mathsf{FUNC}_{\mathcal{X} \to \mathcal{Y}}} [\mathcal{A}^{f(\cdot)}(1^\lambda) = 1] \right|$$

is negligible.

2.6 Entropy-Smoothing Function

Here, we review the definition of an entropy-smoothing function.

Definition 9 (Entropy-Smoothing Function). *Let* $\mathsf{H} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ *be an efficiently computable function. We say that* H *is* entropy-smoothing*, if for any PPT adversary* $\mathcal{A}$*,*

$$\mathsf{Adv}^{\mathtt{ESF}}_{\mathsf{H},\mathcal{A}}(\lambda) := \left| \Pr_{\substack{k \xleftarrow{\$} \mathcal{K} \\ x \xleftarrow{\$} \mathcal{X}}} [\mathcal{A}(k, \mathsf{H}(k,x)) = 1] - \Pr_{\substack{k \xleftarrow{\$} \mathcal{K} \\ y \xleftarrow{\$} \mathcal{Y}}} [\mathcal{A}(k,y) = 1] \right|$$

is negligible.

We note that the above security definition is satisfied by a (strong) randomness extractor (where we view its seed as a key k of an entropy-smoothing function). Hence, an entropy-smoothing function exists unconditionally.

3 Refining Security Definitions of Sender-Anamorphic Encryption

In this section, we provide formal definitions of our refined security notions for SAME that we call `SAME-R` and `S-SAME-R` security. Furthermore, we introduce their variants that we call `SAME-R′` and `S-SAME-R′` security, which do not capture specific threats but are useful for security proofs.

In Sections 3.1 and 3.2, we review the syntax and correctness, and the security definitions, respectively, for ℓ-SAME proposed by Wang et al. [29], which is a generalization of the original model proposed by Persiano et al. [26]. In ℓ-SAME, one duplicate message is transmitted through ℓ ciphertexts of forced messages, and the original definition of Persiano et al. is captured as 1-SAME. After reviewing the existing definitions, we introduce formal definitions of our refined security notions for ℓ-SAME in Section 3.3.

3.1 Syntax and Correctness

Let $\ell \in \mathbb{N}$. An ℓ-SAME scheme AME consists of the six algorithms $(\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{fRand}, \mathsf{dDec})$, among which each of $(\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ is the same as the corresponding algorithm of a PKE scheme. fRand and dDec are the algorithms specific to ℓ-SAME with the following syntax:

$\mathsf{fRand}(\mathsf{pp}, (\mathsf{fpk}_i)_{i\in[\ell]}, (\mathsf{fm}_i)_{i\in[\ell]}, \mathsf{dpk}, \mathsf{dm}) \xrightarrow{\$} (r_i)_{i\in[\ell]}$**:** This is the fake randomness generation algorithm. It takes a public parameter pp, ℓ forced public keys $(\mathsf{fpk}_i)_{i\in[\ell]}$, ℓ forced messages $(\mathsf{fm}_i)_{i\in[\ell]}$, a duplicate public key dpk, and a duplicate message dm as input. It outputs ℓ random values $(r_i)_{i\in[\ell]} \in \mathcal{R}^\ell$, where $\mathcal{R}$ is the randomness space for Enc.

$\mathsf{dDec}(\mathsf{dsk}, (\mathsf{c}_i)_{i\in[\ell]}) \to \mathsf{dm}/\perp$**:** This is the decryption algorithm for anamorphic ciphertexts. It takes a duplicate secret key dsk and ℓ ciphertexts $(\mathsf{c}_i)_{i\in[\ell]}$ as input, and outputs a duplicate plaintext dm, or $\perp$ to indicate a decryption failure.

Correctness. We require that for any $\lambda \in \mathbb{N}$, any set of forced messages $\mathsf{fm}_1, \ldots, \mathsf{fm}_\ell$, and any duplicate message dm, the following probability is negligible:

$$\Pr\left[\begin{array}{ll} \mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda); & \\ \forall i \in [\ell],\ (\mathsf{fpk}_i, \mathsf{fsk}_i) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp}); & \\ (\mathsf{dpk}, \mathsf{dsk}) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp}); & : \mathsf{dDec}(\mathsf{dsk},\ (\mathsf{c}_i)_{i\in[\ell]}) \neq \mathsf{dm} \\ (r_i)_{i\in[\ell]} \xleftarrow{\$} \mathsf{fRand}(\mathsf{pp}, \mathsf{FPK}, \mathsf{FM}, \mathsf{dpk}, \mathsf{dm}); & \\ \forall i \in [\ell],\ \mathsf{c}_i \leftarrow \mathsf{Enc}(\mathsf{fpk}_i,\ \mathsf{fm}_i;\ r_i) & \end{array}\right],$$

where FPK denotes $(\mathsf{fpk}_i)_{i\in[\ell]}$, and FM denotes $(\mathsf{fm}_i)_{i\in[\ell]}$.

3.2 Existing Security Definitions

Here, we review the existing security definitions for ℓ-SAME introduced by Wang et al. [29], which are in turn based on the original security definition for SAME introduced by Persiano et al. [26]. To clearly distinguish these existing definitions from our refined security definitions, we call the existing security definitions "SAME security" and "S-SAME security".[4] The former is the default security notion, while the latter represents a stronger version. Both notions capture indistinguishability in the sense that an adversary holding forced public keys should be unable to determine whether given ciphertexts are normal or anamorphic. In the stronger setting, the adversary knows all the forced secret keys and a duplicate public key in addition to forced public keys, which is the only difference between SAME security and S-SAME security.

Formally, the definitions of SAME and S-SAME security are as follows.

Definition 10 ((Strong) SAME Security). *Let* $\mathtt{XXX} \in \{\mathtt{SAME}, \mathtt{S\text{-}SAME}\}$. *We say that an ℓ-SAME scheme* AME *is* XXX *secure if it satisfies the following conditions 1 and 2:*

1. $(\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ *constitutes an* IND-CPA *secure PKE scheme.*
2. *For any PPT adversary* $\mathcal{A}$, $\mathsf{Adv}^{\mathtt{XXX}}_{\mathsf{AME},\mathcal{A}}(\lambda) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{XXX}}_{\mathsf{AME},\mathcal{A}}(\lambda) = 1] - \frac{1}{2}|$ *is negligible, where the experiment* $\mathsf{Expt}^{\mathtt{XXX}}_{\mathsf{AME},\mathcal{A}}(\lambda)$ *is defined as in Fig. 4.*

Although the stronger version might seem to not capture natural threats, it does capture some realistic situations, for example, when a national authority manages all citizens' keys, while the secret key of a foreign embassy within the country remains outside its control. Moreover, this stronger notion is theoretically significant due to its connections with receiver AME [29].

3.3 Our Refined Security Definitions

Here, we introduce our refined security definition, SAME-R security, and its stronger version, S-SAME-R security, which are security notions for ℓ-SAME where randomness used for encryption is disclosed to an adversary. Specifically, these definitions are obtained by simply modifying the experiments for SAME security and S-SAME security so that the encryption oracle's response includes the randomness used for encryption. Receiving randomness in addition to ciphertexts, an adversary (i.e., a dictator) can verify that the ciphertexts are indeed computed from forced public keys, forced messages, and the randomness. Therefore, our formalization captures the threat that the dictator forces a sender to send ciphertexts as instructed.

[4] Wang et al. [29] also defined the security property called robustness for ℓ-SAME. Its formal definition, as well as a discussion on the robustness of the (ℓ-)SAME schemes treated in this paper, are given in Appendix C.

$\underline{\mathsf{Expt}^{\mathtt{XXX}}_{\mathsf{AME},\mathcal{A}}(\lambda)}$:
$b \xleftarrow{\$} \{0,1\}$
$\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$
for $i \in [\ell]$
$(\mathsf{fpk}_i, \mathsf{fsk}_i) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp})$
$(\mathsf{dpk}, \mathsf{dsk}) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp})$
if $\mathtt{XXX} \in \{\mathtt{S\text{-}SAME}, \mathtt{S\text{-}SAME\text{-}R}, \mathtt{S\text{-}SAME\text{-}R'}\}$ then
$\mathsf{aux} \leftarrow \big((\mathsf{fsk}_i)_{i\in[\ell]}, \mathsf{dpk}\big)$
else
$\mathsf{aux} \leftarrow \emptyset$
$b' \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_b^{\mathtt{XXX}}(\cdot,\cdot)}\big(\mathsf{pp}, (\mathsf{fpk}_i)_{i\in[\ell]}, \mathsf{aux}\big)$
return $(b \stackrel{?}{=} b')$

$\underline{\mathcal{O}_b^{(\mathtt{S\text{-}})\mathtt{SAME}}(\mathsf{FM}, \mathsf{dm})}$:
if $b = 0$ then
$(r_i)_{i\in[\ell]} \xleftarrow{\$} \mathsf{fRand}\big(\mathsf{pp}, \mathsf{FPK}, \mathsf{FM}, \mathsf{dpk}, \mathsf{dm}\big)$
if $b = 1$ then
$(r_i)_{i\in[\ell]} \xleftarrow{\$} \mathcal{R}^\ell$
for $i \in [\ell]$
$\mathsf{c}_i \leftarrow \mathsf{Enc}(\mathsf{fpk}_i, \mathsf{fm}_i; r_i)$
return $(\mathsf{c}_i)_{i\in[\ell]}$

Fig. 4. The experiment for $\mathtt{XXX} \in \{\mathtt{SAME}, \mathtt{S\text{-}SAME}, \mathtt{SAME\text{-}R}, \mathtt{S\text{-}SAME\text{-}R}, \mathtt{SAME\text{-}R'}, \mathtt{S\text{-}SAME\text{-}R'}\}$ security for ℓ-SAME (left), and the procedure of the oracle for SAME and S-SAME security (right). (S-)SAME denotes either SAME or S-SAME, FPK denotes $(\mathsf{fpk}_i)_{i\in[\ell]}$, and FM denotes $(\mathsf{fm}_i)_{i\in[\ell]}$. Procedures of the oracles in the experiments for SAME-R, S-SAME-R, SAME-R′, and S-SAME-R′ security will be given in Fig. 5.

We also introduce variants of our new security notions that we call SAME-R′ and S-SAME-R′, where the oracle's response in the experiments consists only of the randomness, excluding the ciphertext. The usefulness of these variants will be explained shortly after the formal definition.

Formally, the definitions of SAME-R, S-SAME-R, SAME-R′, and S-SAME-R′ security are as follows.

Definition 11 (SAME-R and SAME-R′ Security and Their Stronger Variants). *Let* $\mathtt{XXX} \in \{\mathtt{SAME\text{-}R}, \mathtt{S\text{-}SAME\text{-}R}, \mathtt{SAME\text{-}R'}, \mathtt{S\text{-}SAME\text{-}R'}\}$. *We say that an ℓ-SAME scheme* AME *is* XXX *secure if it satisfies the following conditions 1 and 2:*

1. $(\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ *constitutes an* IND-CPA *secure PKE scheme.*
2. *For any PPT adversary* $\mathcal{A}$, $\mathsf{Adv}^{\mathtt{XXX}}_{\mathsf{AME},\mathcal{A}}(\lambda) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{XXX}}_{\mathsf{AME},\mathcal{A}}(\lambda) = 1] - \frac{1}{2}|$ *is negligible, where the experiment* $\mathsf{Expt}^{\mathtt{XXX}}_{\mathsf{AME},\mathcal{A}}(\lambda)$ *is defined as in Fig. 4 and the procedure of the oracle* $\mathcal{O}_b^{\mathtt{XXX}}$ *is defined as in Fig. 5.*

It should be obvious that SAME-R (resp. S-SAME-R) security implies SAME (resp. S-SAME) security, since an adversary considered in the former gets more information than the latter.

Theorem 1. *If an ℓ-SAME scheme* AME *is* SAME-R *(resp.* S-SAME-R*) secure, then* AME *is also* SAME *(resp.* S-SAME*) secure.*

Next, we show the equivalence between SAME-R (resp. S-SAME-R) and SAME-R′ (resp. S-SAME-R′). By this equivalence, in order to prove SAME-R (resp. S-SAME-R) security, it is sufficient to prove SAME-R′ (resp. S-SAME-R′) security.

$\underline{\mathcal{O}_b^{\mathtt{XXX}}(\mathsf{FM}, \mathsf{dm})}$:
  if $b = 0$ then
    $(r_i)_{i\in[\ell]} \xleftarrow{\$} \mathsf{fRand}\big(\mathsf{pp}, \mathsf{FPK}, \mathsf{FM}, \mathsf{dpk}, \mathsf{dm}\big)$
  if $b = 1$ then
    $(r_i)_{i\in[\ell]} \xleftarrow{\$} \mathcal{R}^\ell$
  for $i \in [\ell]$
    $\mathsf{c}_i \leftarrow \mathsf{Enc}(\mathsf{fpk}_i, \mathsf{fm}_i; r_i)$
  if $\mathtt{XXX} \in \{\mathtt{SAME\text{-}R}, \mathtt{S\text{-}SAME\text{-}R}\}$ then
    return $(\mathsf{c}_i, r_i)_{i\in[\ell]}$
  if $\mathtt{XXX} \in \{\mathtt{SAME\text{-}R'}, \mathtt{S\text{-}SAME\text{-}R'}\}$ then
    return $(r_i)_{i\in[\ell]}$

Fig. 5. Procedure of the oracle in the experiments for SAME-R, S-SAME-R, SAME-R′ and S-SAME-R′ security. FPK denotes $(\mathsf{fpk}_i)_{i\in[\ell]}$, and FM denotes $(\mathsf{fm}_i)_{i\in[\ell]}$. $\mathcal{O}_b^{\mathtt{SAME\text{-}R}}$ and $\mathcal{O}_b^{\mathtt{S\text{-}SAME\text{-}R}}$ are the encryption oracles whose output includes not only ciphertexts but also randomness used for generating the ciphertexts, and $\mathcal{O}_b^{\mathtt{SAME\text{-}R'}}$ and $\mathcal{O}_b^{\mathtt{S\text{-}SAME\text{-}R'}}$ are the randomness oracles that only output randomness used for generating the ciphertexts.

Theorem 2. *An ℓ-SAME scheme* AME *is* SAME-R *(resp.* S-SAME-R*) secure if and only if* AME *is* SAME-R′ *(resp.* S-SAME-R′*) secure.*

The equivalence can be established since the queried forced messages FM are chosen by an adversary itself, and thus if the adversary is given the randomness behind the ciphertexts, then it can compute the ciphertexts by itself. For completeness, we provide the formal proof in Appendix B.

Although these security notions do not capture certain specific threats, their usefulness lies in simplifying security proofs via the above equivalence. In particular, this equivalence allows us to prove SAME-R (resp. S-SAME-R) security by analyzing the simpler notions SAME-R′ (resp. S-SAME-R′) security, focusing only on the randomness of the oracle's responses.

4 Persiano et al.'s SAME Scheme

Persiano et al. [26] showed how a PKE scheme satisfying several additional properties can be turned into a SAME scheme that can encrypt a 1-bit duplicate message. In this section, we investigate whether their scheme satisfies our refined security notions, and show that it satisfies SAME-R security under exactly the same assumption originally used to prove its SAME security, while it does not satisfy S-SAME-R security as there is an easy attack.

We first recall their SAME scheme. Let $\mathsf{PKE} = (\mathsf{Setup}', \mathsf{KG}', \mathsf{Enc}', \mathsf{Dec}')$ be a 1-bit PKE scheme that satisfies the following additional properties.[5]

– *(P1) Common randomness property.* For every two public keys pk_0 and pk_1 output by KG' and for every ciphertext c produced using public key pk_0 and

[5] The description of the properties here is mostly taken verbatim from [26, Section 6.1].

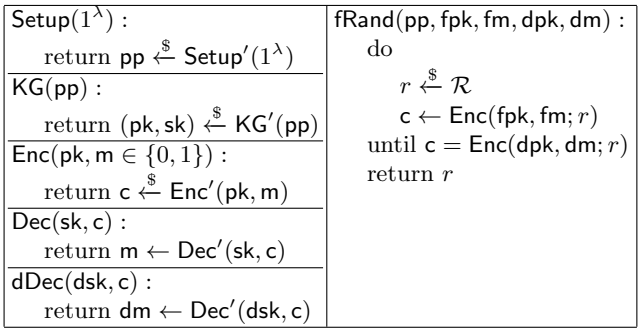

Fig. 6. Persiano et al.'s SAME scheme $\mathsf{AME}_{\mathtt{PPY}}$ constructed from a 1-bit PKE scheme with additional properties (P1) to (P3). In the description, $\mathcal{R}$ denotes the randomness space of $\mathsf{Enc}' = \mathsf{Enc}$.

randomness r, there exists a message m such that $\mathsf{c} = \mathsf{Enc}'(\mathsf{pk}_1, \mathsf{m}; r)$; that is, c is a ciphertext also for public key pk_1 with the randomness r.

- *(P2) Message recovery from randomness.* It is possible to recover the plaintext carried by a ciphertext c from the randomness r used to produce it and the public key pk.
- *(P3) Equal distribution of plaintexts.* All public keys share the same ciphertext space, and for a ciphertext c in the common ciphertext space and for a random key pair $(\mathsf{pk}, \mathsf{sk})$ as sampled by KG', c is the ciphertext of the bit 0 with a probability 1/2.

Based on PKE, Persiano et al.'s SAME scheme $\mathsf{AME}_{\mathtt{PPY}} = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{fRand}, \mathsf{dDec})$ is constructed as in Fig. 6. In the figure, $\mathcal{R}$ denotes the randomness space of $\mathsf{Enc}' = \mathsf{Enc}$.

Persiano et al. [26] showed the following.

Theorem 3. *If* PKE *is* IND-CPA *secure and satisfies the additional properties (P1) to (P3), then* $\mathsf{AME}_{\mathtt{PPY}}$ *is* SAME *secure.*

We now state our results on Persiano et al.'s SAME scheme regarding our refined security notions for SAME.

Theorem 4. *Under exactly the same assumption as used in Theorem 3,* $\mathsf{AME}_{\mathtt{PPY}}$ *is* SAME-R *secure.*

Proof. (sketch) According to the original proof for Theorem 3 in [26], the argument for showing that a genuine ciphertext is indistinguishable from an anamorphic ciphertext, namely, the indistinguishability of a ciphertext $\mathsf{c} = \mathsf{Enc}(\mathsf{fpk}, \mathsf{fm}; r)$ in the cases where

- r is sampled honestly from the randomness space of Enc, and
- r is generated as $r \xleftarrow{\$} \mathsf{fRand}(\mathsf{pp}, \mathsf{fpk}, \mathsf{fm}, \mathsf{dpk}, \mathsf{dm})$,

is purely statistical. Thus, the indistinguishability holds even against computationally unbounded adversaries, and giving the randomness behind the ciphertext to the adversary does no harm to the original argument, which implies that $\mathsf{AME}_{\mathtt{PPY}}$ is $\mathtt{SAME\text{-}R}$ secure. □ (**Theorem 4**)

Theorem 5. $\mathsf{AME}_{\mathtt{PPY}}$ *is not* $\mathtt{S\text{-}SAME\text{-}R}$ *secure. Specifically, there exists a PPT adversary* $\mathcal{A}$ *against the* $\mathtt{S\text{-}SAME\text{-}R}$ *security of* $\mathsf{AME}_{\mathtt{PPY}}$ *such that*

$$\mathsf{Adv}^{\mathtt{S\text{-}SAME\text{-}R}}_{\mathsf{AME}_{\mathtt{PPY}},\mathcal{A}}(\lambda) = \frac{1}{2}.$$

Proof. We first describe an adversary $\mathcal{A}$ that attacks the $\mathtt{S\text{-}SAME\text{-}R}$ security of $\mathsf{AME}_{\mathtt{PPY}}$:

$\mathcal{A}(\mathsf{pp}, (\mathsf{fpk}_i, \mathsf{fsk}_i)_{i\in[\ell]}, \mathsf{dpk})$**:** $\mathcal{A}$ picks arbitrary $\mathsf{fm}, \mathsf{dm} \in \{0,1\}$, and submits $(\mathsf{fm}, \mathsf{dm})$ to the encryption oracle $\mathcal{O}_b^{\mathtt{S\text{-}SAME\text{-}R}}$, which returns a ciphertext/randomness pair (c, r) to $\mathcal{A}$. Then, $\mathcal{A}$ sets $b' \leftarrow (\mathsf{Enc}(\mathsf{dpk}, \mathsf{dm}; r) \stackrel{?}{=} \mathsf{c})$, and terminates with output b'.

Let b be the challenge bit in the $\mathtt{S\text{-}SAME\text{-}R}$ experiment for $\mathcal{A}$. If $b = 0$ (i.e. r is computed using fRand), then it always holds $\mathsf{Enc}(\mathsf{dpk}, \mathsf{dm}; r) = \mathsf{c}$, and thus it holds that $\Pr[b' = 1 | b = 0] = 1$. On the other hand, if $b = 1$ (i.e. r is sampled randomly), then $\mathsf{Enc}(\mathsf{dpk}, \mathsf{dm}; r) = \mathsf{c}$ holds with probability $1/2$ due to the property (P3) (Equal distribution of plaintexts), and thus it holds that $\Pr[b' = 1 | b = 1] = \Pr[b' = 0 | b = 1] = 1/2$. Hence, $\mathcal{A}$'s advantage in attacking the $\mathtt{S\text{-}SAME\text{-}R}$ security of $\mathsf{AME}_{\mathtt{PPY}}$ is

$$\mathsf{Adv}^{\mathtt{S\text{-}SAME\text{-}R}}_{\mathsf{AME}_{\mathtt{PPY}},\mathcal{A}}(\lambda) = \left|\Pr[b' = 1 | b = 0] - \Pr[b' = 1 | b = 1]\right| = \left|1 - \frac{1}{2}\right| = \frac{1}{2},$$

which is clearly non-negligible. □ (**Theorem 5**)

5 Wang et al.'s First SAME Scheme and Its Modified Scheme

In this section, we investigate whether Wang et al.'s first ℓ-SAME scheme satisfies our refined security notions. Specifically, in Section 5.1, we show that their first scheme satisfies $\mathtt{SAME\text{-}R}$ security. Furthermore, we consider a modified version of their scheme by slightly changing the building blocks, and show that the modified scheme satisfies $\mathtt{S\text{-}SAME\text{-}R}$ security. This result and the reason why we consider such a modified scheme will be explained in Section 5.2.

5.1 Wang et al.'s First Scheme and Its $\mathtt{SAME\text{-}R}$ Security

Firstly, we recall Wang et al.'s first ℓ-SAME scheme proposed in [29, Section 4], which uses a pseudorandom PKE scheme and an entropy-smoothing function as building blocks.

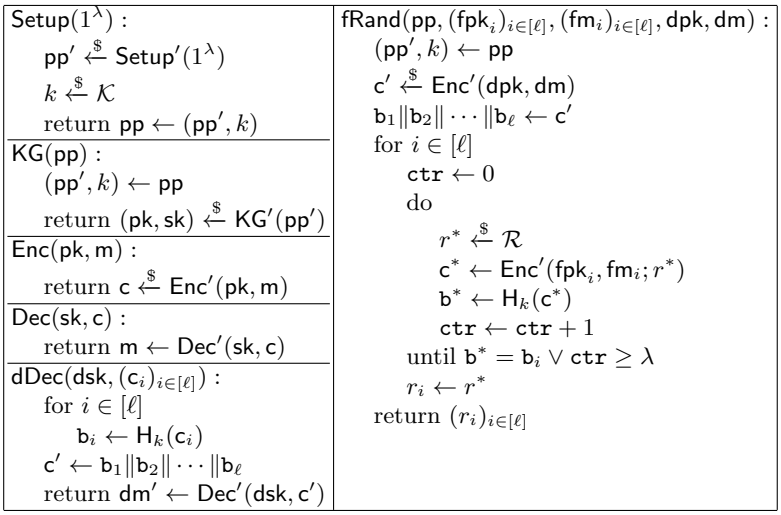

Fig. 7. Wang et al.'s first ℓ-SAME scheme $\mathsf{AME}_{\mathtt{WCHY1}}$ constructed from a pseudorandom PKE scheme and an entropy-smoothing function. In the figure, $\mathtt{b}_i \in \{0,1\}$ for every $i \in [\ell]$.

Formally, let $\mathsf{PKE} = (\mathsf{Setup}', \mathsf{KG}', \mathsf{Enc}', \mathsf{Dec}')$ be a PKE scheme. For simplicity, we assume that the message length encrypted by PKE is fixed to be some polynomial $n = n(\lambda)$. Let $\ell = \ell(\lambda)$ denote the ciphertext length of PKE corresponding to the fixed message length n, and let $\mathsf{H}\colon \mathcal{K} \times \{0,1\}^\ell \to \{0,1\}$ be an entropy-smoothing function where $\mathcal{K}$ denote its key space. (For better readability, we write $\mathsf{H}_k(\cdot)$ to mean $\mathsf{H}(k, \cdot)$.)

Using these building blocks, Wang et al.'s first ℓ-SAME scheme $\mathsf{AME}_{\mathtt{WCHY1}} = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{fRand}, \mathsf{dDec})$ is constructed as in Fig. 7.[6]

Wang et al. showed the following theorems that guarantee the correctness and SAME security of $\mathsf{AME}_{\mathtt{WCHY1}}$.

Theorem 6. *If* PKE *satisfies pseudorandomness and* H *is entropy-smoothing, then* $\mathsf{AME}_{\mathtt{WCHY1}}$ *satisfies correctness.*

Theorem 7. *If* PKE *satisfies pseudorandomness and* H *is entropy-smoothing, then* $\mathsf{AME}_{\mathtt{WCHY1}}$ *satisfies* SAME *security.*

Since the proof for the correctness of $\mathsf{AME}_{\mathtt{WCHY1}}$ is not explicitly given in [29], we provide the formal proof of Theorem 6 in Appendix D.1 for completeness.

[6] In Wang et al.'s original scheme, the counter ctr for the "do-until" loop in the description of fRand is not used, and thus is not guaranteed to terminate. Strictly speaking, therefore, fRand with the original description is an expected polynomial-time algorithm, not a PPT algorithm. In this paper, we introduce the counter so that it terminates in polynomial time.

$\underline{\mathsf{Expt}^{\mathtt{Fail}}_{\mathsf{PKE},\mathsf{H},\mathcal{A}}(\lambda)}$:
$k \xleftarrow{\$} \mathcal{K}$
$\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$
$(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp})$
$\mathsf{m} \xleftarrow{\$} \mathcal{A}(k, \mathsf{pp}, \mathsf{pk})$
$\forall h \in [\lambda+1] : \mathsf{c}_h \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$
$\forall h \in [\lambda+1] : \mathtt{b}_h \leftarrow \mathsf{H}_k(\mathsf{c}_h)$
if $\bigwedge_{h=1}^{\lambda}(\mathtt{b}_h \neq \mathtt{b}_{\lambda+1})$ then
  return 1
else
  return 0

Fig. 8. Experiment for evaluating the probability of the "failure" that could occur in an execution of fRand.

For convenience for our correctness proof (of Theorem 6), and also for our security proof below, let us introduce the following lemma, which can be used to estimate the probability that the "do-until" loop in an execution of fRand ends without finding a good fake randomness.

Lemma 3. *Let* PKE *be a PKE scheme whose plaintext (resp. ciphertext) space is* $\{0,1\}^n$ *(resp.* $\{0,1\}^\ell$*) for some polynomials* $n = n(\lambda)$ *and* $\ell = \ell(\lambda)$*. Let* $\mathsf{H} \colon \mathcal{K} \times \{0,1\}^\ell \to \{0,1\}$ *be an efficiently computable function. If* PKE *satisfies pseudorandomness and* H *satisfies entropy-smoothness, then for any PPT adversary* $\mathcal{A}$*,* $\mathsf{Adv}^{\mathtt{Fail}}_{\mathsf{PKE},\mathsf{H},\mathcal{A}}(\lambda) := \Pr[\mathsf{Expt}^{\mathtt{Fail}}_{\mathsf{PKE},\mathsf{H},\mathcal{A}}(\lambda) = 1]$ *is negligible, where* $\mathsf{Expt}^{\mathtt{Fail}}_{\mathsf{PKE},\mathsf{H},\mathcal{A}}(\lambda)$ *is the experiment defined as in Fig. 8.*

More specifically, for any PPT adversary $\mathcal{A}$*, there exist PPT adversaries* $\mathcal{B}$ *and* $\mathcal{B}'$ *such that* $\mathsf{Adv}^{\mathtt{Fail}}_{\mathsf{PKE},\mathsf{H},\mathcal{A}}(\lambda) \leq \lambda\Big(\mathsf{Adv}^{\mathtt{PR}}_{\mathsf{PKE},\mathcal{B}}(\lambda) + \mathsf{Adv}^{\mathtt{ESF}}_{\mathsf{H},\mathcal{B}'}(\lambda)\Big) + 2^{-\lambda}$.

The formal proof is given in Appendix D.2. An intuition behind the proof is that if the ciphertexts $\{\mathsf{c}_h\}_{h\in[\lambda]}$ are pseudorandom and $\{\mathtt{b}_h\}_{h\in[\lambda]}$ are computed by an entropy-smoothing function H using each of $\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_\lambda$ as input, then the bits $\mathtt{b}_1 \| \mathtt{b}_2 \| \cdots \| \mathtt{b}_\lambda$ behave like uniformly random bits, and thus the probability that none of $\mathtt{b}_1, \mathtt{b}_2, \ldots, \mathtt{b}_\lambda$ hits $\mathtt{b}_{\lambda+1}$ is exponentially small in λ.

We now show that $\mathsf{AME}_{\mathtt{WCHY1}}$ satisfies SAME-R security.

Theorem 8. *If* PKE *satisfies pseudorandomness and* H *is entropy-smoothing, then* $\mathsf{AME}_{\mathtt{WCHY1}}$ *satisfies* SAME-R *security.*

Intuitions behind the Proof. In the proof, we would like to construct a sequence of games in which the oracle's behavior is changed from using randomness generated by fRand to using uniformly sampled randomness.

In the (original) proof for SAME security of $\mathsf{AME}_{\mathtt{WCHY1}}$ in [29], such a transition is made possible by replacing all ciphertexts of forced messages and a duplicate message with random ones. As a result, the decision on whether to use sampled

randomness r in oracle responses no longer depends on the queries of an adversary, and the resulting distribution of output randomness becomes identical to that of true randomness. However, in the SAME-R security model, where the randomness is revealed to the adversary, this approach no longer works. If we attempt to replace ciphertexts of forced messages with random ones using the pseudorandomness of the underlying PKE scheme, the reduction algorithm cannot return r that remains consistent with the embedded instance. Consequently, the simulation fails, and the original proof strategy cannot be applied directly in the SAME-R setting.

To prove SAME-R security, we instead modify the ciphertext of a duplicate message dm rather than the ciphertexts of the forced messages $\mathsf{fm}_1, \ldots, \mathsf{fm}_\ell$. Since the randomness used to generate the ciphertext of dm is not revealed to the adversary, we can exploit this fact together with the pseudorandomness of the underlying PKE scheme. Specifically, we replace the ciphertext of dm with a concatenation of $\mathsf{b}_i = \mathsf{H}_k(\mathsf{Enc}(\mathsf{fpk}_i, \mathsf{fm}_i))$ for every $i \in [\ell]$, using an additional intermediate game in which this ciphertext is sampled uniformly at random. As a result, for every $i \in [\ell]$, the generation process of b_i becomes identical to that of b^* generated in the do-until loop, and by applying the properties of rejection sampling and conditional resampling (Lemmas 1 and 2, respectively), we can show that the resulting randomness distribution is indistinguishable from that of true randomness. The full details are presented in the formal security proof below.

Proof. (of Theorem 8.) Since IND-CPA security of PKE follows from the pseudorandomness of PKE, $(\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ is IND-CPA secure. By Theorem 2, it is sufficient to show that $\mathsf{AME}_{\mathtt{WCHY1}}$ is SAME-R′ secure. Let $\mathcal{A}$ be a PPT adversary against the SAME-R′ security that makes $q = q(\lambda)$ queries to the randomness oracle. We will show that for this $\mathcal{A}$, there exist PPT adversaries $\mathcal{B}, \mathcal{B}', \mathcal{B}''$, and $\mathcal{B}'''$ that satisfy

$$\begin{aligned}&\mathsf{Adv}^{\mathtt{SAME\text{-}R'}}_{\mathsf{AME}_{\mathtt{WCHY1}},\mathcal{A}}(\lambda)\\ &\leq q\mathsf{Adv}^{\mathtt{PR}}_{\mathsf{PKE},\mathcal{B}}(\lambda) + q\ell\Big(\mathsf{Adv}^{\mathtt{ESF}}_{\mathsf{H},\mathcal{B}'}(\lambda) + \mathsf{Adv}^{\mathtt{PR}}_{\mathsf{PKE},\mathcal{B}''}(\lambda) + \mathsf{Adv}^{\mathsf{Fail}}_{\mathsf{PKE},\mathsf{H},\mathcal{B}'''}(\lambda)\Big). \qquad (1)\end{aligned}$$

Note that since we assume PKE satisfies pseudorandomness, and H is entropy smoothing, the advantage $\mathsf{Adv}^{\mathsf{Fail}}_{\mathsf{PKE},\mathsf{H},\mathcal{B}'''}(\lambda)$ for $\mathcal{B}'''$ is negligible by Lemma 3. Hence, $\mathcal{A}$'s advantage is negligible, which implies the theorem.

We show the above via a sequence of games argument using eight games, described in the following. Since the only difference in $\mathsf{Expt}^{\mathtt{SAME\text{-}R'}}_{\mathsf{AME}_{\mathtt{WCHY1}},\mathcal{A}}(\lambda)$ between the cases in which the challenge bit 0 and 1 lies in the randomness oracle, we will focus on its behavior. For convenience, Figs. 9 and 10 describe the behaviors of the randomness oracle in each game. Let b be the challenge bit and b' be the guess bit output by $\mathcal{A}$ in $\mathsf{Expt}^{\mathtt{SAME\text{-}R'}}_{\mathsf{AME}_{\mathtt{WCHY1}},\mathcal{A}}(\lambda)$. Furthermore, for $N \in \{0, \ldots, 7\}$, let $\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game}N}$ be the randomness oracle in Game N, and let G_N be the event that $\mathcal{A}$ returns 1 in Game $N \in \{0, 1, \ldots, 7\}$.

Game 0: This is exactly $\mathsf{Expt}^{\mathtt{SAME\text{-}R'}}_{\mathsf{AME}_{\mathtt{WCHY1}},\mathcal{A}}(\lambda)$ in which the challenge bit $b = 0$.

$\underline{\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game0}}((\mathsf{fm}_i)_{i\in[\ell]},\mathsf{dm}):}$
$\mathsf{c} \xleftarrow{\$} \mathsf{Enc}'(\mathsf{dpk},\mathsf{dm})$
$\mathtt{b}_1\|\mathtt{b}_2\|\cdots\|\mathtt{b}_\ell \leftarrow \mathsf{c}$
for $i \in [\ell]$
  $\mathtt{ctr} \leftarrow 0$
  do
    $r^* \xleftarrow{\$} \mathcal{R}$
    $\mathsf{c}^* \leftarrow \mathsf{Enc}'(\mathsf{fpk}_i,\mathsf{fm}_i;r^*)$
    $\mathtt{b}^* \leftarrow \mathsf{H}_k(\mathsf{c}^*)$
    $\mathtt{ctr} \leftarrow \mathtt{ctr}+1$
  until $\mathtt{b}^* = \mathtt{b}_i \vee \mathtt{ctr} \geq \lambda$
  $r_i \leftarrow r^*$
return $(r_i)_{i\in[\ell]}$

$\underline{\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game1}}((\mathsf{fm}_i)_{i\in[\ell]},\mathsf{dm}):}$
$\mathsf{c} \xleftarrow{\$} \{0,1\}^\ell$
$\mathtt{b}_1\|\mathtt{b}_2\|\cdots\|\mathtt{b}_\ell \leftarrow \mathsf{c}$
for $i \in [\ell]$
  $\mathtt{ctr} \leftarrow 0$
  do
    $r^* \xleftarrow{\$} \mathcal{R}$
    $\mathsf{c}^* \leftarrow \mathsf{Enc}'(\mathsf{fpk}_i,\mathsf{fm}_i;r^*)$
    $\mathtt{b}^* \leftarrow \mathsf{H}_k(\mathsf{c}^*)$
    $\mathtt{ctr} \leftarrow \mathtt{ctr}+1$
  until $\mathtt{b}^* = \mathtt{b}_i \vee \mathtt{ctr} \geq \lambda$
  $r_i \leftarrow r^*$
return $(r_i)_{i\in[\ell]}$

$\underline{\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game2}}((\mathsf{fm}_i)_{i\in[\ell]},\mathsf{dm}):}$
for $i \in [\ell]$
  $\mathtt{b}_i \xleftarrow{\$} \{0,1\}$
  $\mathtt{ctr} \leftarrow 0$
  do
    $r^* \xleftarrow{\$} \mathcal{R}$
    $\mathsf{c}^* \leftarrow \mathsf{Enc}'(\mathsf{fpk}_i,\mathsf{fm}_i;r^*)$
    $\mathtt{b}^* \leftarrow \mathsf{H}_k(\mathsf{c}^*)$
    $\mathtt{ctr} \leftarrow \mathtt{ctr}+1$
  until $\mathtt{b}^* = \mathtt{b}_i \vee \mathtt{ctr} \geq \lambda$
  $r_i \leftarrow r^*$
return $(r_i)_{i\in[\ell]}$

$\underline{\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game3}}((\mathsf{fm}_i)_{i\in[\ell]},\mathsf{dm}):}$
for $i \in [\ell]$
  $\widehat{\mathsf{c}}_i \xleftarrow{\$} \{0,1\}^\ell$
  $\mathtt{b}_i \leftarrow \mathsf{H}_k(\widehat{\mathsf{c}}_i)$
  $\mathtt{ctr} \leftarrow 0$
  do
    $r^* \xleftarrow{\$} \mathcal{R}$
    $\mathsf{c}^* \leftarrow \mathsf{Enc}'(\mathsf{fpk}_i,\mathsf{fm}_i;r^*)$
    $\mathtt{b}^* \leftarrow \mathsf{H}_k(\mathsf{c}^*)$
    $\mathtt{ctr} \leftarrow \mathtt{ctr}+1$
  until $\mathtt{b}^* = \mathtt{b}_i \vee \mathtt{ctr} \geq \lambda$
  $r_i \leftarrow r^*$
return $(r_i)_{i\in[\ell]}$

Fig. 9. Behaviors of the randomness oracle $\mathcal{O}^{\mathsf{Rand}}$ in Games 0 through 3 in the proof of Theorem 8.

Game 1: This game is the same as Game 0 except for how c is generated. Specifically, $\mathsf{c} \in \{0,1\}^\ell$ is chosen uniformly at random instead of running Enc'.

Game 2: This game is the same as Game 1 except for how $\mathtt{b}_1, \mathtt{b}_2, \ldots, \mathtt{b}_\ell$ are generated. Specifically, instead of sampling $\mathtt{b}_1||\mathtt{b}_2||\cdots||\mathtt{b}_\ell$ at once, we sample it bit by bit uniformly at random inside the "for" loop.

Game 3: This game is the same as Game 2 except for how $\mathtt{b}_i$ is generated inside the "for" loop for each $i \in [\ell]$. Specifically, for every $i \in [\ell]$, $\widehat{\mathsf{c}}_i \in \{0,1\}^\ell$ is chosen uniformly at random and $\mathtt{b}_i \leftarrow \mathsf{H}_k(\widehat{\mathsf{c}}_i)$ is computed, instead of randomly sampling $\mathtt{b}_i$.

Game 4: This game is the same as Game 3 except for how $\widehat{\mathsf{c}}_i$ is computed inside the "for" loop for each $i \in [\ell]$. Specifically, for each $i \in [\ell]$, $\widehat{r}_i \in \mathcal{R}$ is chosen uniformly at random and $\widehat{\mathsf{c}}_i$ is computed by $\widehat{\mathsf{c}}_i \leftarrow \mathsf{Enc}'(\mathsf{fpk}_i,\mathsf{fm}_i;\widehat{r}_i)$, instead of sampling a random ℓ-bit string.

$\underline{\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game4}}((\mathsf{fm}_i)_{i\in[\ell]},\mathsf{dm}):}$
for $i \in [\ell]$
$\widehat{r}_i \xleftarrow{\$} \mathcal{R}$
$\widehat{\mathsf{c}}_i \leftarrow \mathsf{Enc}'(\mathsf{fpk}_i, \mathsf{fm}_i;\ \widehat{r}_i)$
$\mathsf{b}_i \leftarrow \mathsf{H}_k(\widehat{\mathsf{c}}_i)$
$\mathtt{ctr} \leftarrow 0$
do
$r^* \xleftarrow{\$} \mathcal{R}$
$\mathsf{c}^* \leftarrow \mathsf{Enc}'(\mathsf{fpk}_i, \mathsf{fm}_i; r^*)$
$\mathsf{b}^* \leftarrow \mathsf{H}_k(\mathsf{c}^*)$
$\mathtt{ctr} \leftarrow \mathtt{ctr} + 1$
until $\mathsf{b}^* = \mathsf{b}_i \vee \mathtt{ctr} \geq \lambda$
$r_i \leftarrow r^*$
return $(r_i)_{i\in[\ell]}$

$\underline{\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game5}}((\mathsf{fm}_i)_{i\in[\ell]},\mathsf{dm}):}$
for $i \in [\ell]$
$\widehat{r}_i \xleftarrow{\$} \mathcal{R}$
$\widehat{\mathsf{c}}_i \leftarrow \mathsf{Enc}'(\mathsf{fpk}_i, \mathsf{fm}_i; \widehat{r}_i)$
$\mathsf{b}_i \leftarrow \mathsf{H}_k(\widehat{\mathsf{c}}_i)$
do
$r^* \xleftarrow{\$} \mathcal{R}$
$\mathsf{c}^* \xleftarrow{\$} \mathsf{Enc}'(\mathsf{fpk}_i, \mathsf{fm}_i; r^*)$
$\mathsf{b}^* \leftarrow \mathsf{H}_k(\mathsf{c}^*)$
until $\mathsf{b}^* = \mathsf{b}_i$
$r_i \leftarrow r^*$
return $(r_i)_{i\in[\ell]}$

$\underline{\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game6}}((\mathsf{fm}_i)_{i\in[\ell]},\mathsf{dm}):}$
for $i \in [\ell]$
$\widehat{r}_i \xleftarrow{\$} \mathcal{R}$
$\widehat{\mathsf{c}}_i \leftarrow \mathsf{Enc}'(\mathsf{fpk}_i, \mathsf{fm}_i;\ \widehat{r}_i)$
$\mathsf{b}_i \leftarrow \mathsf{H}_k(\widehat{\mathsf{c}}_i)$
define:
$\mathcal{R}_i := \left\{ r \in \mathcal{R} \;\middle|\; \begin{array}{l} \mathsf{Enc}'(\mathsf{fpk}_i, \mathsf{fm}_i; r) = \mathsf{c}_i \\ \wedge\ \mathsf{H}_k(\mathsf{c}_i) = \mathsf{b}_i \end{array} \right\}$
$r_i \xleftarrow{\$} \mathcal{R}_i$
return $(r_i)_{i\in[\ell]}$

$\underline{\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game7}}((\mathsf{fm}_i)_{i\in[\ell]},\mathsf{dm}):}$
for $i \in [\ell]$
$r_i \xleftarrow{\$} \mathcal{R}$
return $(r_i)_{i\in[\ell]}$

Fig. 10. Behaviors of the randomness oracle $\mathcal{O}^{\mathsf{Rand}}$ in Games 4 through 7 in the proof of Theorem 8.

Game 5: This game is the same as Game 4 except for the processes and the condition regarding $\mathtt{ctr}$. Specifically, in the randomness oracle, the operations $\mathtt{ctr} \leftarrow 0$ and $\mathtt{ctr} \leftarrow \mathtt{ctr} + 1$ are removed, and the loop-termination condition is changed from "until $\mathsf{b}^* = \mathsf{b}_i \vee \mathtt{ctr} \geq \lambda$" to "until $\mathsf{b}^* = \mathsf{b}_i$".

Game 6: This game is the same as Game 5 except for how $(r_i)_{i\in[\ell]}$ are generated. Specifically, for every $i \in [\ell]$, r_i is sampled uniformly at random from

$$\mathcal{R}_i := \left\{ r \in \mathcal{R} \;\middle|\; \mathsf{Enc}'(\mathsf{fpk}_i, \mathsf{fm}_i; r) = \mathsf{c}_i \wedge \mathsf{H}_k(\mathsf{c}_i) = \mathsf{b}_i \right\}.$$

Game 7: In this game, for every $i \in [\ell]$, r_i is chosen uniformly at random from $\mathcal{R}$, and $(r_i)_{i\in[\ell]}$ is returned as the answer to $\mathcal{A}$'s query to the randomness oracle.

Let G_N be the event that $\mathcal{A}$ returns 1 in Game $N \in \{0, 1, \ldots, 7\}$. By the definition of Game 0, it holds that $\Pr[b' = 1 \mid b = 0] = \Pr[\mathsf{G}_0]$. Furthermore, since Game 7 is exactly $\mathsf{Expt}^{\mathtt{SAME\text{-}R'}}_{\mathsf{AME}_{\mathtt{WCHY1}},\mathcal{A}}(\lambda)$ in which the challenge bit $b = 1$, it

holds that $\Pr\left[b' = 1 \mid b = 1\right] = \Pr[\mathsf{G}_7]$. Then, by the definition of the events and the triangle inequality, we have

$$\begin{aligned}\mathsf{Adv}^{\mathtt{SAME\text{-}R'}}_{\mathsf{AME}_{\mathtt{WCHY1}},\mathcal{A}}(\lambda) &= \left|\Pr\left[b' = 1 \mid b = 1\right] - \Pr\left[b' = 1 \mid b = 0\right]\right| \\ &= \left|\Pr[\mathsf{G}_0] - \Pr[\mathsf{G}_7]\right| \\ &\leq \sum_{i=0}^{6}\left|\Pr[\mathsf{G}_i] - \Pr[\mathsf{G}_{i+1}]\right|. \end{aligned} \tag{2}$$

Due to Eq. (2) and the following Lemmas 4 to 10, we can conclude that there exist PPT adversaries $\mathcal{B}, \mathcal{B}', \mathcal{B}''$, and $\mathcal{B}'''$ that satisfy Eq. (1).

Lemma 4. *There exists a PPT adversary $\mathcal{B}$ such that*

$$\left|\Pr[\mathsf{G}_0] - \Pr[\mathsf{G}_1]\right| = q\mathsf{Adv}^{\mathtt{PR}}_{\mathsf{PKE},\mathcal{B}}(\lambda).$$

The proof of this lemma is given in Appendix D.3.

Lemma 5. $\Pr[\mathsf{G}_1] = \Pr[\mathsf{G}_2]$ *holds.*

Proof. This is true since an ℓ-bit string chosen uniformly at random has the same distribution as an ℓ-bit string formed by concatenating random 1-bit strings. □ (**Lemma 5**)

Lemma 6. *There exists a PPT adversary $\mathcal{B}'$ such that*

$$\left|\Pr[\mathsf{G}_2] - \Pr[\mathsf{G}_3]\right| = q\ell\mathsf{Adv}^{\mathtt{ESF}}_{\mathsf{H},\mathcal{B}'}(\lambda).$$

The proof of this lemma is given in Appendix D.4.

Lemma 7. *There exists a PPT adversary $\mathcal{B}''$ such that*

$$\left|\Pr[\mathsf{G}_3] - \Pr[\mathsf{G}_4]\right| = q\ell\mathsf{Adv}^{\mathtt{PR}}_{\mathsf{PKE},\mathcal{B}''}(\lambda).$$

The proof of this lemma is given in Appendix D.5.

Lemma 8. *There exists a PPT adversary $\mathcal{B}'''$ such that*

$$\left|\Pr[\mathsf{G}_4] - \Pr[\mathsf{G}_5]\right| \leq q\ell\mathsf{Adv}^{\mathtt{Fail}}_{\mathsf{PKE},\mathsf{H},\mathcal{B}'''}(\lambda).$$

Proof. For $s \in [q]$, let $\mathsf{Bad}_{4,s}$ be the event in Game 4 that, in the process of generating a response to the s-th query during the game, the "do-until" loop for some $i \in [\ell]$ terminates because the counter $\mathtt{ctr}$ reaches λ before the condition $\mathsf{b}^* = \mathsf{b}_i$ is satisfied. Formally, let $\mathsf{Bad}_{4,s}$ be the event that the condition "$\mathsf{b}^* \neq \mathsf{b}_i \ \wedge \ \mathtt{ctr} \geq \lambda$" holds in some i-th iteration of the "for" loop for s-th query. Even in Game 5, $\mathsf{Bad}_{5,s}$ can be defined in the same manner by conceptually maintaining $\mathtt{ctr}$ without using it for any conditional evaluation. Hence, it is easy to see that

$$\Pr[\mathsf{Bad}_{4,s}] = \Pr[\mathsf{Bad}_{5,s}].$$

$\mathsf{Bad}_{4,s}$ and $\mathsf{Bad}_{5,s}$ can occur with the same probability, and if the bad event does not occur, the two games, Game 4 and Game 5, behave identically. Therefore, it holds

$$\Pr[\mathsf{G}_4 \wedge \bigwedge_{s\in[q]} \overline{\mathsf{Bad}_{4,s}}] = \Pr[\mathsf{G}_5 \wedge \bigwedge_{s\in[q]} \overline{\mathsf{Bad}_{4,s}}].$$

Hence, we have

$$\begin{aligned}\Big|\Pr[\mathsf{G}_4] - \Pr[\mathsf{G}_5]\Big| &= \Bigg|\Big(\Pr\Big[\mathsf{G}_4 \wedge \bigvee_{s\in[q]} \mathsf{Bad}_{4,s}\Big] + \Pr\Big[\mathsf{G}_4 \wedge \bigwedge_{s\in[q]} \overline{\mathsf{Bad}_{4,s}}\Big]\Big) \\ &\qquad - \Big(\Pr\Big[\mathsf{G}_5 \wedge \bigvee_{s\in[q]} \mathsf{Bad}_{5,s}\Big] + \Pr\Big[\mathsf{G}_5 \wedge \bigwedge_{s\in[q]} \overline{\mathsf{Bad}_{5,s}}\Big]\Big)\Bigg| \\ &\leq \sum_{s\in[q]} \Pr[\mathsf{Bad}_{4,s}].\end{aligned}$$

Furthermore, for $s \in [q]$ and $i \in [\ell]$, let $\mathsf{Bad}_{4,s,i}$ be the specific event in Game 4 that "$\mathsf{b}^* \neq \mathsf{b}_i \wedge \mathtt{ctr} \geq \lambda$" holds in the i-th iteration of the "for" loop for the s-th query. By the definition of $\mathsf{Bad}_{4,s,i}$, we have

$$\begin{aligned}\Pr[\mathsf{Bad}_{4,s}] &= \Pr[\mathsf{Bad}_{4,s,1} \vee \mathsf{Bad}_{4,s,2} \vee \cdots \vee \mathsf{Bad}_{4,s,\ell}] \\ &\leq \sum_{i\in[\ell]} \Pr[\mathsf{Bad}_{4,s,i}].\end{aligned}$$

Here, we will show that there exists a PPT adversary $\mathcal{B}'''$ such that

$$\sum_{s\in[q]} \sum_{i\in[\ell]} \Pr[\mathsf{Bad}_{4,s,i}] = q\ell \mathsf{Adv}^{\mathtt{Fail}}_{\mathsf{PKE},\mathsf{H},\mathcal{B}'''}(\lambda).$$

Using $\mathcal{A}$ as a building block, we construct a reduction algorithm $\mathcal{B}'''$ in Lemma 3, which, on input $(k, \mathsf{pp}, \mathsf{pk})$, picks $t \in [q]$ and $j \in [\ell]$ uniformly at random, sets $\mathsf{pp}' \leftarrow (\mathsf{pp}, k)$, generates $(\mathsf{fpk}_i, \mathsf{fsk}_i) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp}')$ for every $i \in [\ell]\backslash\{j\}$, sets $\mathsf{fpk}_j \leftarrow \mathsf{pk}$, and runs $\mathcal{A}(\mathsf{pp}', (\mathsf{fpk}_i)_{i\in[\ell]})$. For $\mathcal{A}$'s s-th query $((\mathsf{fm}_i)_{i\in[\ell]}, \mathsf{dm})$ to the oracle where $s < t$, $\mathcal{B}'''$ executes the procedure of the randomness oracle and returns its result. When $\mathcal{B}'''$ receives the t-th oracle access from $\mathcal{A}$, $\mathcal{B}'''$ submits fm_j to the challenger and terminates the game. $\mathcal{B}'''$ simulates the game up to the t-th

oracle access perfectly. If $\mathsf{Expt}^{\mathtt{Fail}}_{\mathsf{PKE},\mathsf{H},\mathcal{B}'''}(\lambda) = 1$, then the event $\mathsf{Bad}_{\mathtt{SAME},s,i}$ occurs. Specifically, this implies that in the j-th iteration of the "for" loop during the s-th oracle query, the condition $\mathsf{b}^* \neq \mathsf{b}_i$ persists until the counter reaches λ. Therefore, we have

$$\begin{aligned}\mathsf{Adv}^{\mathtt{Fail}}_{\mathsf{PKE},\mathsf{H},\mathcal{B}'''}(\lambda) &= \Pr[\mathsf{Expt}^{\mathtt{Fail}}_{\mathsf{PKE},\mathsf{H},\mathcal{B}'''}(\lambda) = 1] \\ &= \sum_{s\in[q]}\sum_{i\in[\ell]} \Pr[t = s \wedge j = i] \cdot \Pr[\mathsf{Bad}_{4,s,i}] \\ &= \frac{1}{q\ell}\sum_{s\in[q]}\sum_{i\in[\ell]} \Pr[\mathsf{Bad}_{4,s,i}].\end{aligned}$$

Hence, it holds that

$$\left|\Pr[\mathsf{G}_4] - \Pr[\mathsf{G}_5]\right| \leq q\ell\mathsf{Adv}^{\mathtt{Fail}}_{\mathsf{PKE},\mathsf{H},\mathcal{B}'''}(\lambda),$$

which completes the proof. □ (**Lemma 8**)

Lemma 9. $\Pr[\mathsf{G}_5] = \Pr[\mathsf{G}_6]$ *holds.*

Proof. Fix $\mathcal{A}$'s query, $i \in [n]$, and b_i, and let us focus on the distribution of r_i. Let P be the predicate defined by $\mathsf{P}(r) := (\mathsf{H}_k(\mathsf{Enc}(\mathsf{fpk}_i, \mathsf{fm}_i; r)) \stackrel{?}{=} \mathsf{b}_i)$. Then, it is straightforward to see that the set $\mathcal{R}_i$ defined in Game 6 is exactly $\mathsf{P}^{-1}(1)$, from which r_i is chosen uniformly at random in Game 6. On the other hand, in Game 5, the distribution of r_i can be represented as follows:

$$\left\{h \leftarrow 0;\ \text{do } \{h \leftarrow h+1; r_h \stackrel{\$}{\leftarrow} \mathcal{R}\} \text{ until } \mathsf{P}(r_h) = 1 : r_h\right\}.$$

Hence, invoking Lemma 1 with the above predicate P, we can see that the distribution of r_i in Game 6 is identical to the distribution of r_i in Game 5, which implies $\Pr[\mathsf{G}_5] = \Pr[\mathsf{G}_6]$ as desired. □ (**Lemma 9**)

Lemma 10. $\Pr[\mathsf{G}_6] = \Pr[\mathsf{G}_7]$ *holds.*

Proof. In Game 7, for every $i \in [\ell]$, the distribution of r_i is uniform over $\mathcal{R}$.

Let f be the function defined by $f(r) := \mathsf{H}_k(\mathsf{Enc}(\mathsf{fpk}_i, \mathsf{fm}_i; r))$. In Game 6, for every $i \in [\ell]$, the distribution of r_i is denoted as follows:

$$\begin{aligned}&\left\{\begin{array}{l}\widehat{r} \stackrel{\$}{\leftarrow} \mathcal{R}; \\ \mathsf{b}_i \leftarrow \mathsf{H}_k(\mathsf{Enc}(\mathsf{fpk}_i, \mathsf{fm}_i; \widehat{r})); \\ r_i \stackrel{\$}{\leftarrow} \left\{r \in \mathcal{R} \;\middle|\; \mathsf{H}_k(\mathsf{Enc}(\mathsf{fpk}_i, \mathsf{fm}_i; r)) = \mathsf{b}_i\right\}\end{array} : r_i\right\} \\ &\equiv \left\{\widehat{r} \stackrel{\$}{\leftarrow} \mathcal{R}; \mathsf{b}_i \leftarrow f(\widehat{r}); r_i \stackrel{\$}{\leftarrow} f^{-1}(\mathsf{b}_i) : r_i\right\}\end{aligned}$$

For every $i \in [\ell]$, by Lemma 2 with $f(r) := \mathsf{H}_k(\mathsf{Enc}(\mathsf{fpk}_i, \mathsf{fm}_i; r))$, $\{r \stackrel{\$}{\leftarrow} \mathcal{R} : r\} \equiv \{\widehat{r} \stackrel{\$}{\leftarrow} \mathcal{R}; \mathsf{b}_i \leftarrow f(\widehat{r}); r_i \stackrel{\$}{\leftarrow} f^{-1}(\mathsf{b}_i) : r_i\}$ holds. Therefore, we can see that r_i has the same distribution in Games 6 and 7, and thus we have $\Pr[\mathsf{G}_6] = \Pr[\mathsf{G}_7]$. □ (**Lemma 10**)

This completes the proof. □ (**Theorem 8**)

5.2 Modifying Wang et al.'s First Scheme for S-SAME-R Security

In this subsection, we propose a modified version of Wang et al.'s first ℓ-SAME scheme $\mathsf{AME}_{\mathtt{WCHY1}}$, and prove its S-SAME-R security.

Difficulty in the Original Scheme and Our Ideas behind the Modified Scheme. Unfortunately, we have not been able to prove or disprove that $\mathsf{AME}_{\mathtt{WCHY1}}$ satisfies the stronger notion of S-SAME-R security. Let us explain the difficulty. In the proof of SAME-R security shown in the previous subsection, the transition to the final game where true randomness is returned from the randomness oracle was made possible by ensuring that the generation process of b_i for every $i \in [\ell]$ is identical to that of b^* in the "do-until" loop in fRand. Then, the properties of the rejection sampling (Lemma 1) and conditional resampling (Lemma 2) enable us to replace the fake randomness with truly random values.

However, in the stronger security setting of S-SAME-R, the same strategy no longer works. Recall that an adversary in the S-SAME-R experiment needs to be provided the forced secret keys $\{\mathsf{fsk}_i\}_{i\in[\ell]}$ as input. However, when we attempt to make the generation method of b identical inside and outside the "do-until" loop by replacing $\widehat{\mathsf{c}}_i \xleftarrow{\$} \{0,1\}^\ell$ with $\widehat{\mathsf{c}}_i \leftarrow \mathsf{Enc}'(\mathsf{fpk}_i, \mathsf{fm}_i; \widehat{r}_i)$ (for a randomly chosen $\widehat{r}_i$), as in the game transition from Game 3 to Game 4 in the proof of Theorem 8, the reduction algorithm against the pseudorandomness of PKE cannot provide the S-SAME-R adversary with fsk_i in which the challenge instance should be embedded. Thus, the simulation cannot be carried out. Although alternative approaches might exist, we have not been able to find an effective method.

The main difficulty arises from the fact that the inputs to the entropy-smoothing function H cannot be made random, and thus it seems that the property of H cannot be properly used. To overcome the difficulty, a natural idea is to assume that H is a PRF. Then, we could expect that H behaves like a random function even when its inputs are not uniformly random. However, since the key k of H is known to the adversary in the construction of $\mathsf{AME}_{\mathtt{WCHY1}}$, it seems that we cannot also rely on the security of the PRF, since the inputs H is dependent on the adversary's queries. However, if we can guarantee that the inputs to H is independent of the adversary's queries, there is a hope that we can rely on the PRF-security of H.

To make the above ideas work, we consider a modified version of $\mathsf{AME}_{\mathtt{WCHY1}}$. Specifically, we now assume that H is a PRF, and that the underlying PKE scheme is hybrid encryption constructed from a KEM and a DEM, and we use only KEM ciphertexts as inputs to H in an execution of fRand. This modification ensures that the generation of $\{\mathsf{b}_i\}$ no longer depends on the adversary's queries, allowing the PRF-security of H and the pseudorandomness of the KEM to be invoked even when k and $\{\mathsf{fsk}_i\}_{i\in[\ell]}$ are known to the adversary. The remaining steps of the security proof can go similarly to those in the proof of Theorem 8.

Formal Description of Our Modified Scheme and Its S-SAME-R *Security.* Let $\mathsf{KEM} = (\mathsf{KSetup}, \mathsf{KKG}, \mathsf{Encap}, \mathsf{Decap})$ be a KEM scheme whose ciphertext length is $\ell_{\mathsf{KEM}} = \ell_{\mathsf{KEM}}(\lambda)$, session-key length is λ, and randomness space is $\mathcal{R}_{\mathsf{KEM}}$.

Let $\mathsf{DEM} = (\mathsf{DEnc}, \mathsf{DDec})$ be a DEM scheme whose message length is fixed to $n = n(\lambda)$, and whose ciphertext length is $\ell_{\mathsf{DEM}} = \ell_{\mathsf{DEM}}(\lambda)$. Let $\ell := \ell_{\mathsf{KEM}} + \ell_{\mathsf{DEM}}$. Furthermore, let $\mathsf{H} : \mathcal{K} \times \{0,1\}^{\ell_{\mathsf{KEM}}} \to \{0,1\}$ be a PRF. (For simplicity, we write $\mathsf{H}_k(\cdot)$ to mean $\mathsf{H}(k, \cdot)$.)

Using these building blocks, our modified version of Wang et al.'s ℓ-SAME scheme $\mathsf{AME}_{\texttt{M-WCHY1}} = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{fRand}, \mathsf{dDec})$ is constructed as in Fig. 11.

$\mathsf{Setup}(1^\lambda)$:
  $\mathsf{pp}' \xleftarrow{\$} \mathsf{KSetup}(1^\lambda)$
  $k \xleftarrow{\$} \mathcal{K}$
  return $\mathsf{pp} \leftarrow (\mathsf{pp}', k)$

$\mathsf{Enc}(\mathsf{pk}, \mathsf{m})$:
  $(\mathsf{c}_{\mathsf{KEM}}, \mathsf{K}) \xleftarrow{\$} \mathsf{Encap}(\mathsf{pk})$
  $\mathsf{c}_{\mathsf{DEM}} \leftarrow \mathsf{DEnc}(\mathsf{K}, \mathsf{m})$
  return $(\mathsf{c}_{\mathsf{KEM}}, \mathsf{c}_{\mathsf{DEM}})$

$\mathsf{Dec}(\mathsf{sk}, \mathsf{c})$:
  $(\mathsf{c}_{\mathsf{KEM}}, \mathsf{c}_{\mathsf{DEM}}) \leftarrow \mathsf{c}$
  $\mathsf{K} \leftarrow \mathsf{Decap}(\mathsf{sk}, \mathsf{c}_{\mathsf{KEM}})$
  if $\mathsf{K} = \bot$ then
    return $\bot$
  else
    return $\mathsf{m} \leftarrow \mathsf{DDec}(\mathsf{K}, \mathsf{c}_{\mathsf{DEM}})$

$\mathsf{dDec}(\mathsf{dsk}, (\mathsf{c}_i)_{i\in[\ell]})$:
  for $i \in [\ell]$
    $(\mathsf{c}_{\mathsf{KEM},i}, \mathsf{c}_{\mathsf{DEM},i}) \leftarrow \mathsf{c}_i$
    $\mathtt{b}_i \leftarrow \mathsf{H}_k(\mathsf{c}_{\mathsf{KEM},i})$
  $\mathsf{c}' \leftarrow \mathtt{b}_1 \| \mathtt{b}_2 \| \cdots \| \mathtt{b}_\ell$
  $(\mathsf{c}'_{\mathsf{KEM}}, \mathsf{c}'_{\mathsf{DEM}}) \leftarrow \mathsf{c}'$
  $\mathsf{K}' \leftarrow \mathsf{Decap}(\mathsf{dsk}, \mathsf{c}'_{\mathsf{KEM}})$
  if $\mathsf{K}' = \bot$ then
    return $\bot$
  else
    return $\mathsf{dm}' \leftarrow \mathsf{DDec}(\mathsf{K}', \mathsf{c}'_{\mathsf{DEM}})$

$\mathsf{KG}(\mathsf{pp})$:
  $(\mathsf{pp}', k) \leftarrow \mathsf{pp}$
  $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{KKG}(\mathsf{pp}')$
  return $(\mathsf{pk}, \mathsf{sk})$

$\mathsf{fRand}(\mathsf{pp}, (\mathsf{fpk}_i)_{i\in[\ell]}, (\mathsf{fm}_i)_{i\in[\ell]}, \mathsf{dpk}, \mathsf{dm})$:
  $(\mathsf{pp}', k) \leftarrow \mathsf{pp}$
  $(\mathsf{c}_{\mathsf{KEM}}, \mathsf{K}) \xleftarrow{\$} \mathsf{Encap}(\mathsf{dpk})$
  $\mathsf{c}_{\mathsf{DEM}} \leftarrow \mathsf{DEnc}(\mathsf{K}, \mathsf{dm})$
  $\mathtt{b}_1 \| \mathtt{b}_2 \| \cdots \| \mathtt{b}_\ell \leftarrow (\mathsf{c}_{\mathsf{KEM}}, \mathsf{c}_{\mathsf{DEM}})$
  for $i \in [\ell]$
    $\mathtt{ctr} \leftarrow 0$
    do
      $r^* \xleftarrow{\$} \mathcal{R}_{\mathsf{KEM}}$
      $(\mathsf{c}^*_{\mathsf{KEM}}, \mathsf{K}^*) \leftarrow \mathsf{Encap}(\mathsf{fpk}_i; r^*)$
      $\mathtt{b}^* \leftarrow \mathsf{H}_k(\mathsf{c}^*_{\mathsf{KEM}})$
      $\mathtt{ctr} \leftarrow \mathtt{ctr} + 1$
    until $\mathtt{b}^* = \mathtt{b}_i \vee \mathtt{ctr} \geq \lambda$
    $r_i \leftarrow r^*$
  return $(r_i)_{i\in[\ell]}$

Fig. 11. Modified version of Wang et al.'s first ℓ-SAME scheme $\mathsf{AME}_{\texttt{M-WCHY1}}$ constructed from a pseudorandom KEM scheme, a pseudorandom DEM scheme, and a PRF. In the figure, $\mathtt{b}_i \in \{0,1\}$ for every $i \in [\ell]$.

For proving the correctness and $\texttt{S-SAME-R}$ security of $\mathsf{AME}_{\texttt{M-WCHY1}}$, it is useful to introduce the following lemma.

Lemma 11. *If* KEM *satisfies ciphertext pseudorandomness and* H *is a PRF, then there exist PPT adversaries* $\mathcal{B}$ *and* $\mathcal{B}'$ *such that*

$$\epsilon := \Pr\left[\begin{array}{ll} k \xleftarrow{\$} \mathcal{K}; & \\ \mathsf{pp}' \xleftarrow{\$} \mathsf{KSetup}(1^\lambda); & \\ (\mathsf{pk},\mathsf{sk}) \xleftarrow{\$} \mathsf{KKG}(\mathsf{pp}'); & : \forall h \in [\lambda], \mathsf{b}_h \neq \mathsf{b}_{\lambda+1} \\ \forall h \in [\lambda+1], (\mathsf{c}_h, \mathsf{K}_h) \xleftarrow{\$} \mathsf{Encap}(\mathsf{pk}); & \\ \forall h \in [\lambda+1], \mathsf{b}_h \leftarrow \mathsf{H}_k(\mathsf{c}_h) & \end{array}\right]$$
$$\leq (\lambda+1)\mathsf{Adv}^{\mathtt{CTX\text{-}PR}}_{\mathsf{KEM},\mathcal{B}}(\lambda) + \mathsf{Adv}^{\mathtt{PRF}}_{\mathsf{H},\mathcal{B}'}(\lambda) + \lambda(\lambda+1)\cdot 2^{-(\ell_{\mathsf{KEM}}+1)} + 2^{-\lambda}.$$

The formal proof is given in Appendix D.6. Other than relying on the PRF-security of H (instead of entropy-smoothing), its proof is analogous to the proof of Lemma 3. (In fact, the proof is somewhat simpler in the sense that there is no adversary in this lemma.)

We now show the formal statements for its correctness and security.

Theorem 9. *If* KEM *satisfies ciphertext pseudorandomness and* H *is a PRF, then* $\mathsf{AME}_{\mathtt{M\text{-}WCHY1}}$ *satisfies correctness.*

Proof. Let $(\mathsf{fm}_i)_{i\in[\ell]} \in (\{0,1\}^n)^\ell$ be ℓ forced messages, and let $\mathsf{dm} \in \{0,1\}^n$ be a duplicate message. Let $\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$, let $(\mathsf{fpk}_i, \mathsf{fsk}_i)_{i\in[\ell]}$ and $(\mathsf{dpk}, \mathsf{dsk})$ be generated by $\mathsf{KG}(\mathsf{pp})$, let $(r_i)_{i\in[\ell]} \xleftarrow{\$} \mathsf{fRand}(\mathsf{pp}, \mathsf{FPK}, \mathsf{FM}, \mathsf{dpk}, \mathsf{dm})$, and let $\mathsf{c}_i \leftarrow \mathsf{Enc}(\mathsf{fpk}_i,\ \mathsf{fm}_i;\ r_i)$ for every $i \in [\ell]$. We will show that $\Pr[\mathsf{dDec}(\mathsf{dsk},\ (\mathsf{c}_i)_{i\in[\ell]}) \neq \mathsf{dm}]$ is negligible.

To show this, we show that $\Pr[\mathsf{dDec}(\mathsf{dsk},\ (\mathsf{c}_i)_{i\in[\ell]}) \neq \mathsf{dm}] \leq \Pr[\mathsf{Bad}] \leq \mathsf{negl}(\lambda)$, where Bad is defined as follows. Let Bad be the event that, in the process of running $\mathsf{fRand}(\mathsf{pp}, (\mathsf{fpk}_i)_{i\in[\ell]}, (\mathsf{fm}_i)_{i\in[\ell]}, \mathsf{dpk}, \mathsf{dm})$, the "do-until" loop for some $i \in [\ell]$ terminates because the counter $\mathtt{ctr}$ reaches λ before the condition $\mathsf{b}^* = \mathsf{b}_i$ is satisfied. Formally, let Bad be the event that the condition "$\mathsf{b}^* \neq \mathsf{b}_i \ \wedge\ \mathtt{ctr} \geq \lambda$" holds in some i-th iteration of the "for" loop for some $i \in [\ell]$.

If Bad does not occur, $\mathsf{b}_1||\mathsf{b}_2||\cdots||\mathsf{b}_\ell$ is an encryption of dm under dpk where $\mathsf{b}_i = \mathsf{H}_k(\mathsf{Encap}(\mathsf{fpk}_i, ; r_i))$ for every $i \in [\ell]$. Hence, $\Pr[\mathsf{dDec}(\mathsf{dsk},\ (\mathsf{c}_i)_{i\in[\ell]}) \neq \mathsf{dm} \wedge \overline{\mathsf{Bad}}] = 0$ holds by the correctness of KEM and DEM since dDec is executed by Decap and DDec. Therefore, we have $\Pr[\mathsf{dDec}(\mathsf{dsk},\ (\mathsf{c}_i)_{i\in[\ell]}) \neq \mathsf{dm}] \leq \Pr[\mathsf{Bad}]$.

It remains to show that $\Pr[\mathsf{Bad}]$ is negligible. The situation that $\mathsf{b}^* \neq \mathsf{b}_i \wedge \mathtt{ctr} \geq \lambda$ occurs in the execution inside the "do-until" loop of fRand is exactly the same as the event for ϵ defined in Lemma 11. Hence, by the union bound (over the positions $i \in [\ell]$), we have $\Pr[\mathsf{Bad}] \leq \ell\epsilon$, which is negligible by Lemma 11, the ciphertext pseudorandomness of KEM, and the PRF-security of H.

□ (**Theorem 9**)

Theorem 10. *If* KEM *satisfies pseudorandomness,* DEM *satisfies pseudorandomness, and* H *is a PRF, then* $\mathsf{AME}_{\texttt{M-WCHY1}}$ *satisfies* S-SAME-R *security.*

The proof is based on the ideas explained at the beginning of this subsection. In its proof, the following lemma plays the key role, which allows us to overcome the difficulties that exist in the original scheme $\mathsf{AME}_{\texttt{WCHY1}}$. Using this lemma, we can still conduct the transitions of the games, despite in the situation where the S-SAME-R adversary is given the PRF-key k and the forced secret keys $\{\mathsf{fsk}_i\}_{i\in[\ell]}$ as input.

Lemma 12. *Define the distributions* D_{real} *and* D_{rand} *as follows:*

$$D_{\mathsf{real}} := \left\{ \begin{array}{l} k \leftarrow \mathcal{K}; \\ \mathsf{pp}' \xleftarrow{\$} \mathsf{KSetup}(1^\lambda); \\ (\mathsf{pk},\mathsf{sk}) \xleftarrow{\$} \mathsf{KKG}(\mathsf{pp}'); \\ r \xleftarrow{\$} \mathcal{R}_{\mathsf{KEM}}; \\ (\mathsf{c}_{\mathsf{KEM}},\mathsf{K}) \leftarrow \mathsf{Encap}(\mathsf{pk};r) \end{array} : \Big(\mathsf{pp}', k, \mathsf{pk}, \mathsf{sk}, \mathsf{H}_k(\mathsf{c}_{\mathsf{KEM}})\Big) \right\},$$

$$D_{\mathsf{rand}} := \left\{ \begin{array}{l} k \xleftarrow{\$} \mathcal{K}; \\ \mathsf{pp}' \leftarrow \mathsf{KSetup}(1^\lambda); \\ (\mathsf{pk},\mathsf{sk}) \xleftarrow{\$} \mathsf{KKG}(\mathsf{pp}'); \\ y \xleftarrow{\$} \{0,1\} \end{array} : \Big(\mathsf{pp}', k, \mathsf{pk}, \mathsf{sk}, y\Big) \right\}.$$

Then, there exist PPT adversaries $\mathcal{A}$ *and* $\mathcal{A}'$ *such that*

$$\mathbf{SD}(D_{\mathsf{real}}, D_{\mathsf{rand}}) \leq \frac{1}{\sqrt{2}} \left(\mathsf{Adv}^{\mathsf{PRF}}_{\mathsf{H},\mathcal{A}}(\lambda) + \mathsf{Adv}^{\texttt{CTX-PR}}_{\mathsf{KEM},\mathcal{A}'}(\lambda) + 2^{-\ell_{\mathsf{KEM}}} \right)^{\frac{1}{2}}.$$

In particular, if KEM *satisfies ciphertext pseudorandomness and* H *is a PRF, then* $\mathbf{SD}(D_{\mathsf{real}}, D_{\mathsf{rand}})$ *is negligible.*

The formal proof of Lemma 12 is given in Appendix D.7. Its proof is somewhat analogous to the proof for the leftover hash lemma [19].

Proof. (of Theorem 10). IND-CPA security of $(\mathsf{Setup},\mathsf{KG},\mathsf{Enc},\mathsf{Dec})$ follows from the pseudorandomness of KEM and DEM. By Theorem 2, it is sufficient to show that $\mathsf{AME}_{\texttt{M-WCHY1}}$ is S-SAME-R′ secure. Let $\mathcal{A}$ be a PPT adversary against the S-SAME-R′ security that makes q queries to the randomness oracle. We will show that for this $\mathcal{A}$, there exist PPT adversaries $\mathcal{B}, \mathcal{B}', \mathcal{B}''_0, \mathcal{B}''_1, \mathcal{B}'''_0$, and $\mathcal{B}'''_1$ that satisfy

$$\begin{aligned} &\mathsf{Adv}^{\texttt{S-SAME-R}'}_{\mathsf{AME}_{\texttt{M-WCHY1}},\mathcal{A}}(\lambda) \\ &\leq q\Big(\mathsf{Adv}^{\mathsf{PR}}_{\mathsf{KEM},\mathcal{B}}(\lambda) + \mathsf{Adv}^{\mathsf{PR}}_{\mathsf{DEM},\mathcal{B}'}(\lambda)\Big) \\ &\quad + \frac{q\ell}{\sqrt{2}} \left(\mathsf{Adv}^{\mathsf{PRF}}_{\mathsf{H},\mathcal{B}''_0}(\lambda) + \mathsf{Adv}^{\mathsf{PR}}_{\mathsf{KEM},\mathcal{B}''_1}(\lambda) + 2^{-\ell_{\mathsf{KEM}}} \right)^{\frac{1}{2}} \\ &\quad + q\ell\Big((\lambda+1)\mathsf{Adv}^{\texttt{CTX-PR}}_{\mathsf{KEM},\mathcal{B}'''_0}(\lambda) + \mathsf{Adv}^{\mathsf{PRF}}_{\mathsf{H},\mathcal{B}'''_1}(\lambda) + \lambda(\lambda+1)\cdot 2^{-(\ell+1)} + 2^{-\lambda}\Big). \quad (3) \end{aligned}$$

$\underline{\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game0}}((\mathsf{fm}_i)_{i\in[\ell]}, \mathsf{dm}):}$
$(\mathsf{c}_{\mathsf{KEM}}, \mathsf{K}) \xleftarrow{\$} \mathsf{Encap}(\mathsf{dpk})$
$\mathsf{c}_{\mathsf{DEM}} \leftarrow \mathsf{DEnc}(\mathsf{K}, \mathsf{dm})$
$\mathtt{b}_1 \| \mathtt{b}_2 \| \cdots \| \mathtt{b}_\ell \leftarrow (\mathsf{c}_{\mathsf{KEM}}, \mathsf{c}_{\mathsf{DEM}})$
for $i \in [\ell]$
  $\mathtt{ctr} \leftarrow 0$
  do
    $r^* \xleftarrow{\$} \mathcal{R}_{\mathsf{KEM}}$
    $(\mathsf{c}^*_{\mathsf{KEM}}, \mathsf{K}^*) \leftarrow \mathsf{Encap}(\mathsf{fpk}_i; r^*)$
    $\mathtt{b}^* \leftarrow \mathsf{H}_k(\mathsf{c}^*_{\mathsf{KEM}})$
    $\mathtt{ctr} \leftarrow \mathtt{ctr} + 1$
  until $\mathtt{b}^* = \mathtt{b}_i \vee \mathtt{ctr} \geq \lambda$
  $r_i \leftarrow r^*$
return $(r_i)_{i\in[\ell]}$

$\underline{\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game1}}((\mathsf{fm}_i)_{i\in[\ell]}, \mathsf{dm}):}$
$\mathsf{c}_{\mathsf{KEM}} \xleftarrow{\$} \{0,1\}^{\ell_{\mathsf{KEM}}}$
$\mathsf{K} \xleftarrow{\$} \{0,1\}^{\lambda}$
$\mathsf{c}_{\mathsf{DEM}} \leftarrow \mathsf{DEnc}(\mathsf{K}, \mathsf{dm})$
$\mathtt{b}_1 \| \mathtt{b}_2 \| \cdots \| \mathtt{b}_\ell \leftarrow (\mathsf{c}_{\mathsf{KEM}}, \mathsf{c}_{\mathsf{DEM}})$
for $i \in [\ell]$
  $\mathtt{ctr} \leftarrow 0$
  do
    $r^* \xleftarrow{\$} \mathcal{R}_{\mathsf{KEM}}$
    $(\mathsf{c}^*_{\mathsf{KEM}}, \mathsf{K}^*) \leftarrow \mathsf{Encap}(\mathsf{fpk}_i; r^*)$
    $\mathtt{b}^* \leftarrow \mathsf{H}_k(\mathsf{c}^*_{\mathsf{KEM}})$
    $\mathtt{ctr} \leftarrow \mathtt{ctr} + 1$
  until $\mathtt{b}^* = \mathtt{b}_i \vee \mathtt{ctr} \geq \lambda$
  $r_i \leftarrow r^*$
return $(r_i)_{i\in[\ell]}$

$\underline{\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game2}}((\mathsf{fm}_i)_{i\in[\ell]}, \mathsf{dm}):}$
$\mathsf{c} \xleftarrow{\$} \{0,1\}^{\ell}$
$\mathtt{b}_1 \| \mathtt{b}_2 \| \cdots \| \mathtt{b}_\ell \leftarrow \mathsf{c}$
for $i \in [\ell]$
  $\mathtt{ctr} \leftarrow 0$
  do
    $r^* \xleftarrow{\$} \mathcal{R}_{\mathsf{KEM}}$
    $(\mathsf{c}^*_{\mathsf{KEM}}, \mathsf{K}^*) \leftarrow \mathsf{Encap}(\mathsf{fpk}_i; r^*)$
    $\mathtt{b}^* \leftarrow \mathsf{H}_k(\mathsf{c}^*_{\mathsf{KEM}})$
    $\mathtt{ctr} \leftarrow \mathtt{ctr} + 1$
  until $\mathtt{b}^* = \mathtt{b}_i \vee \mathtt{ctr} \geq \lambda$
  $r_i \leftarrow r^*$
return $(r_i)_{i\in[\ell]}$

$\underline{\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game3}}((\mathsf{fm}_i)_{i\in[\ell]}, \mathsf{dm}):}$
for $i \in [\ell]$
  $\mathtt{b}_i \xleftarrow{\$} \{0,1\}$
  $\mathtt{ctr} \leftarrow 0$
  do
    $r^* \xleftarrow{\$} \mathcal{R}_{\mathsf{KEM}}$
    $(\mathsf{c}^*_{\mathsf{KEM}}, \mathsf{K}^*) \leftarrow \mathsf{Encap}(\mathsf{fpk}_i; r^*)$
    $\mathtt{b}^* \leftarrow \mathsf{H}_k(\mathsf{c}^*_{\mathsf{KEM}})$
    $\mathtt{ctr} \leftarrow \mathtt{ctr} + 1$
  until $\mathtt{b}^* = \mathtt{b}_i \vee \mathtt{ctr} \geq \lambda$
  $r_i \leftarrow r^*$
return $(r_i)_{i\in[\ell]}$

Fig. 12. Behaviors of the randomness oracle in Games 0 through 3 used in the proof of Theorem 10.

Note that since we assume KEM satisfies pseudorandomness, KEM also satisfies ciphertext pseudorandomness. Hence, $\mathcal{A}$'s advantage is negligible, which implies the theorem.

We show the above via a sequence of games argument using eight games, described in the following. Since the only difference in $\mathsf{Expt}^{\mathtt{S\text{-}SAME\text{-}R'}}_{\mathsf{AME}_{\mathtt{M\text{-}WCHY1}},\mathcal{A}}(\lambda)$ between the cases in which the challenge bit 0 and 1 lies in the randomness oracle, we will focus on its behavior. For convenience, Figs. 12 and 13 describe the behaviors of the randomness oracle in each game. Let b be the challenge bit and b' be the guess bit output by $\mathcal{A}$ in $\mathsf{Expt}^{\mathtt{S\text{-}SAME\text{-}R'}}_{\mathsf{AME}_{\mathtt{M\text{-}WCHY1}},\mathcal{A}}(\lambda)$. Furthermore, for $N \in \{0,\ldots,7\}$, let $\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game}N}$ be the randomness oracle in Game N, and let G_N be the event that $\mathcal{A}$ returns 1 in Game $N \in \{0,1,\ldots,7\}$.

Game 0: This is exactly $\mathsf{Expt}^{\mathtt{S\text{-}SAME\text{-}R'}}_{\mathsf{AME}_{\mathtt{M\text{-}WCHY1}},\mathcal{A}}(\lambda)$ in which its challenge bit $b = 0$.

$\underline{\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game4}}((\mathsf{fm}_i)_{i\in[\ell]},\mathsf{dm}):}$
for $i \in [\ell]$
$\widehat{r}_i \xleftarrow{\$} \mathcal{R}_{\mathsf{KEM}}$
$(\widehat{\mathsf{c}}_{\mathsf{KEM},i}, \widehat{\mathsf{K}}_i) \leftarrow \mathsf{Encap}(\mathsf{fpk}_i; \widehat{r}_i)$
$\mathtt{b}_i \leftarrow \mathsf{H}_k(\widehat{\mathsf{c}}_{\mathsf{KEM},i})$
$\mathtt{ctr} \leftarrow 0$
do
$r^* \xleftarrow{\$} \mathcal{R}_{\mathsf{KEM}}$
$(\mathsf{c}^*_{\mathsf{KEM}}, \mathsf{K}^*) \leftarrow \mathsf{Encap}(\mathsf{fpk}_i; r^*)$
$\mathtt{b}^* \leftarrow \mathsf{H}_k(\mathsf{c}^*_{\mathsf{KEM}})$
$\mathtt{ctr} \leftarrow \mathtt{ctr} + 1$
until $\mathtt{b}^* = \mathtt{b}_i \vee \mathtt{ctr} \geq \lambda$
$r_i \leftarrow r^*$
return $(r_i)_{i\in[\ell]}$

$\underline{\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game5}}((\mathsf{fm}_i)_{i\in[\ell]},\mathsf{dm}):}$
for $i \in [\ell]$
$\widehat{r}_i \xleftarrow{\$} \mathcal{R}_{\mathsf{KEM}}$
$(\widehat{\mathsf{c}}_{\mathsf{KEM},i}, \widehat{\mathsf{K}}_i) \leftarrow \mathsf{Encap}(\mathsf{fpk}_i; \widehat{r}_i)$
$\mathtt{b}_i \leftarrow \mathsf{H}_k(\widehat{\mathsf{c}}_{\mathsf{KEM},i})$
do
$r^* \xleftarrow{\$} \mathcal{R}_{\mathsf{KEM}}$
$(\mathsf{c}^*_{\mathsf{KEM}}, \mathsf{K}^*) \leftarrow \mathsf{Encap}(\mathsf{fpk}_i; r^*)$
$\mathtt{b}^* \leftarrow \mathsf{H}_k(\mathsf{c}^*_{\mathsf{KEM}})$
until $\mathtt{b}^* = \mathtt{b}_i$
$r_i \leftarrow r^*$
return $(r_i)_{i\in[\ell]}$

$\underline{\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game6}}((\mathsf{fm}_i)_{i\in[\ell]},\mathsf{dm}):}$
for $i \in [\ell]$
$\widehat{r}_i \xleftarrow{\$} \mathcal{R}_{\mathsf{KEM}}$
$(\widehat{\mathsf{c}}_{\mathsf{KEM},i}, \widehat{\mathsf{K}}_i) \leftarrow \mathsf{Encap}(\mathsf{fpk}_i; \widehat{r}_i)$
$\mathtt{b}_i \leftarrow \mathsf{H}_k(\widehat{\mathsf{c}}_{\mathsf{KEM},i})$
define:
$\mathcal{R}_i := \left\{ r \in \mathcal{R}_{\mathsf{KEM}} \;\middle|\; \begin{array}{l} \mathsf{Encap}(\mathsf{fpk}_i; r) = (\mathsf{c}_{\mathsf{KEM},i}, *) \\ \wedge\ \mathsf{H}_k(\mathsf{c}_{\mathsf{KEM},i}) = \mathtt{b}_i \end{array} \right\}$
$r_i \xleftarrow{\$} \mathcal{R}_i$
return $(r_i)_{i\in[\ell]}$

$\underline{\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game7}}((\mathsf{fm}_i)_{i\in[\ell]},\mathsf{dm}):}$
for $i \in [\ell]$
$r_i \xleftarrow{\$} \mathcal{R}_{\mathsf{KEM}}$
return $(r_i)_{i\in[\ell]}$

Fig. 13. Behaviors of the randomness oracle in Games 4 through 7 used in the proof of Theorem 10.

Game 1: This game is the same as Game 0 except for how $\mathsf{c}_{\mathsf{KEM}}$ and K are generated in the randomness oracle. Specifically, $\mathsf{c}_{\mathsf{KEM}} \in \{0,1\}^{\ell_{\mathsf{KEM}}}$ and $\mathsf{K} \in \{0,1\}^{\lambda}$ are chosen uniformly at random, respectively, instead of running Encap.

Game 2: This game is the same as Game 1 except for how $\mathsf{c}_{\mathsf{DEM}}$ is generated in the randomness oracle. Specifically, $\mathsf{c}_{\mathsf{DEM}} \in \{0,1\}^{\ell_{\mathsf{DEM}}}$ is chosen uniformly at random instead of running DEnc.

Game 3: This game is the same as Game 2 except for how $\mathtt{b}_1, \mathtt{b}_2, \ldots, \mathtt{b}_\ell$ are generated. Specifically, instead of sampling $\mathtt{b}_1||\mathtt{b}_2||\cdots||\mathtt{b}_\ell$ at once, we sample it bit by bit uniformly at random inside the "for" loop.

Game 4: This game is the same as Game 3 except for how $\mathtt{b}_i$ is generated inside the "for" loop. Specifically, in the randomness oracle, the challenger picks $\widehat{r}_i \in \mathcal{R}_{\mathsf{KEM}}$, and computes $(\widehat{\mathsf{c}}_{\mathsf{KEM},i}, \mathsf{K}_i) \leftarrow \mathsf{Encap}(\mathsf{fpk}_i; \widehat{r}_i)$ and $\mathtt{b}_i \leftarrow \mathsf{H}_k(\widehat{\mathsf{c}}_{\mathsf{KEM},i})$ for every $i \in [\ell]$ inside the "for" loop.

Game 5: This game is the same as Game 4 except for the process and the condition regarding $\mathtt{ctr}$. Specifically, in the randomness oracle, the operations $\mathtt{ctr} \leftarrow 0$ and $\mathtt{ctr} \leftarrow \mathtt{ctr} + 1$ are removed, and the loop-termination condition is changed from "until $\mathtt{b}^* = \mathtt{b}_i \vee \mathtt{ctr} \geq \lambda$" to "until $\mathtt{b}^* = \mathtt{b}_i$".

Game 6: This game is the same as Game 5 except for how $(r_i)_{i\in[\ell]}$ are generated. Specifically, for every $i \in [\ell]$, r_i is sampled uniformly at random from

$$\mathcal{R}_i := \Big\{ r \in \mathcal{R}_{\mathsf{KEM}} \;\Big|\; \mathsf{Encap}(\mathsf{fpk}_i; r) = (\mathsf{c}_{\mathsf{KEM},i}, *) \wedge \mathsf{H}_k(\mathsf{c}_{\mathsf{KEM},i}) = \mathtt{b}_i \Big\}.$$

Game 7: In this game, for every $i \in [\ell]$, r_i is chosen uniformly at random from $\mathcal{R}_{\mathsf{KEM}}$, and $(r_i)_{i\in[\ell]}$ is returned as its output.

Let G_N be the event that $\mathcal{A}$ returns 1 in Game $N \in \{0, 1, \ldots, 7\}$. By the definition of Game 0, it holds that $\Pr[b' = 1 \mid b = 0] = \Pr[\mathsf{G}_0]$. Furthermore, since Game 7 is exactly $\mathsf{Expt}^{\mathtt{S\text{-}SAME\text{-}R'}}_{\mathsf{AME}_{\mathtt{M\text{-}WCHY1}},\mathcal{A}}(\lambda)$ in which the challenge bit $b = 1$, it holds that $\Pr\big[b' = 1 \mid b = 1\big] = \Pr[\mathsf{G}_7]$. Then, by the definition of the events and the triangle inequality, we have

$$\begin{aligned}
\mathsf{Adv}^{\mathtt{S\text{-}SAME\text{-}R'}}_{\mathsf{AME}_{\mathtt{M\text{-}WCHY1}},\mathcal{A}}(\lambda) &= \Big|\Pr\big[b' = 1 \mid b = 1\big] - \Pr\big[b' = 1 \mid b = 0\big]\Big| \\
&= \Big|\Pr[\mathsf{G}_0] - \Pr[\mathsf{G}_7]\Big| \\
&\leq \sum_{i=0}^{6} \Big|\Pr[\mathsf{G}_i] - \Pr[\mathsf{G}_{i+1}]\Big|. \qquad (4)
\end{aligned}$$

Due to Eq. (4) and the following Lemmas 13 to 19, we can conclude that there exist PPT adversaries $\mathcal{B}, \mathcal{B}', \mathcal{B}''_0, \mathcal{B}''_1, \mathcal{B}'''_0$, and $\mathcal{B}'''_1$ satisfying Eq. (3).

Lemma 13. *There exists a PPT adversary $\mathcal{B}$ such that*

$$\Big|\Pr[\mathsf{G}_0] - \Pr[\mathsf{G}_1]\Big| = q\mathsf{Adv}^{\mathtt{PR}}_{\mathsf{KEM},\mathcal{B}}(\lambda).$$

The proof of this lemma is given in Appendix D.8.

Lemma 14. *There exists a PPT adversary $\mathcal{B}'$ such that*

$$\Big|\Pr[\mathsf{G}_1] - \Pr[\mathsf{G}_2]\Big| = q\mathsf{Adv}^{\mathtt{PR}}_{\mathsf{DEM},\mathcal{B}'}(\lambda).$$

The proof of this lemma is given in Appendix D.9.

Lemma 15. $\Pr[\mathsf{G}_2] = \Pr[\mathsf{G}_3]$ *holds.*

Proof. This is true since an ℓ-bit string chosen uniformly at random has the same distribution as an ℓ-bit string formed by concatenating random 1-bit strings. □ (**Lemma 15**)

Lemma 16. *There exist PPT adversaries $\mathcal{B}_0''$ and $\mathcal{B}_1''$ such that*

$$\left|\Pr[\mathsf{G}_3] - \Pr[\mathsf{G}_4]\right| \leq \frac{q\ell}{\sqrt{2}}\left(\mathsf{Adv}^{\mathsf{PRF}}_{\mathsf{H},\mathcal{B}_0''}(\lambda) + \mathsf{Adv}^{\mathsf{PR}}_{\mathsf{KEM},\mathcal{B}_1''}(\lambda) + 2^{-\ell_{\mathsf{KEM}}}\right)^{\frac{1}{2}}.$$

Proof. For proving this lemma, we will rely on Lemma 12.

Using $\mathcal{A}$ as a building block, we construct a PPT distinguisher $\mathcal{B}^*$ between D_{real} and D_{rand} defined in Lemma 12, as follows:

$\mathcal{B}^*(\mathsf{pp}', k, \mathsf{pk}, \mathsf{sk}, y)$**:** $\mathcal{B}^*$ picks $t \in [q]$ and $j \in [\ell]$ uniformly at random, and sets $\mathsf{pp} \leftarrow (\mathsf{pp}', k)$ and $(\mathsf{fpk}_j, \mathsf{fsk}_j) \leftarrow (\mathsf{pk}, \mathsf{sk})$. Furthermore, $\mathcal{B}^*$ generates $(\mathsf{fpk}_i, \mathsf{fsk}_i) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp})$ for every $i \in [\ell]\backslash\{j\}$ and $(\mathsf{dpk}, \mathsf{dsk}) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp})$. $\mathcal{B}^*$ runs $\mathcal{A}(\mathsf{pp}, (\mathsf{fpk}_i, \mathsf{fsk}_i)_{i\in[\ell]}, \mathsf{dpk})$ and executes the following procedure for answering $\mathcal{A}$'s s-th query $((\mathsf{fm}_i)_{i\in[\ell]}, \mathsf{dm})$ to the randomness oracle where $s \in [q]$:

- If $s < t$, then $\mathcal{B}^*$ computes $(r_i)_{i\in[\ell]}$ in the same way as $\mathcal{O}^{\mathtt{Rand}}_{\mathsf{Game4}}$, and returns $(r_i)_{i\in[\ell]}$ to $\mathcal{A}$.
- If $s = t$, then $\mathcal{B}^*$ executes the following procedure and returns its result $(r_i)_{i\in[\ell]}$ to $\mathcal{A}$:

for $i \in [\ell]$
  if $i \leq j-1$ then
    $\widehat{r}_i \xleftarrow{\$} \mathcal{R}_{\mathsf{KEM}}$
    $(\widehat{\mathsf{c}}_{\mathsf{KEM},i}, \widehat{\mathsf{K}}_i) \leftarrow \mathsf{Encap}(\mathsf{fpk}_i; \widehat{r}_i)$
    $\mathtt{b}_i \leftarrow \mathsf{H}_k(\widehat{\mathsf{c}}_{\mathsf{KEM},i})$
  else if $i = j$ then
    $\mathtt{b}_j \leftarrow y$
  else if $i \geq j+1$ then
    $\mathtt{b}_i \xleftarrow{\$} \{0,1\}$
  $\mathtt{ctr} \leftarrow 0$
  do
    $r^* \xleftarrow{\$} \mathcal{R}_{\mathsf{KEM}}$
    $(\mathsf{c}^*_{\mathsf{KEM}}, \mathsf{K}^*) \leftarrow \mathsf{Encap}(\mathsf{fpk}_i; r^*)$
    $\mathtt{b}^* \leftarrow \mathsf{H}_k(\mathsf{c}^*_{\mathsf{KEM}})$
    $\mathtt{ctr} \leftarrow \mathtt{ctr} + 1$
  until $\mathtt{b}^* = \mathtt{b}_i \vee \mathtt{ctr} \geq \lambda$
  $r_i \leftarrow r^*$

- If $s > t$, then $\mathcal{B}^*$ computes $(r_i)_{i\in[\ell]}$ in the same way as $\mathcal{O}^{\mathtt{Rand}}_{\mathsf{Game3}}$, and returns $(r_i)_{i\in[\ell]}$ to $\mathcal{A}$.

Finally, $\mathcal{A}$ returns a bit b' and $\mathcal{B}^*$ returns b' as its own guess bit.

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

- If $\mathcal{B}^*$'s input y is sampled from D_{rand} (and y is a random value), $t = 1$, and $j = 1$, then $\mathcal{B}^*$ simulates Game 3 perfectly for $\mathcal{A}$. Therefore, the probability that $\mathcal{B}^*$ returns 1 is identical to the probability that $\mathcal{A}$ returns 1 in Game 3, namely,

$$\Pr_{D_{\mathsf{rand}}}[\mathcal{B}^*(\mathsf{pp}', k, \mathsf{pk}, \mathsf{sk}, y) = 1 \mid t = 1 \wedge j = 1] = \Pr[\mathsf{G}_3]. \tag{5}$$

- If $\mathcal{B}^*$'s input is sampled from D_{real} (and y is an output of H on input a KEM-ciphertext $\mathsf{c}_{\mathsf{KEM}}$), $t = q$, and $j = \ell$, then $\mathcal{B}^*$ simulates Game 4 perfectly for $\mathcal{A}$. Therefore, the probability that $\mathcal{B}^*$ returns 1 is identical to the probability that $\mathcal{A}$ returns 1 in Game 4, namely,

$$\Pr_{D_{\mathsf{real}}}[\mathcal{B}^*(\mathsf{pp}', k, \mathsf{pk}, \mathsf{sk}, \mathsf{H}_k(\mathsf{c}_{\mathsf{KEM}})) = 1 \mid t = q \wedge j = \ell] = \Pr[\mathsf{G}_4]. \tag{6}$$

- From $\mathcal{A}$'s viewpoint, for every $s \in [q]$ and $i \in [\ell - 1]$, the experiments simulated by $\mathcal{B}^*$ in each of the following situations are identical:
  - $\mathcal{B}^*$'s input y is an output of H, $t = s$, and $j = i$.
  - $\mathcal{B}^*$'s input y is a random value, $t = s$, and $j = i + 1$.

  Since $\mathcal{B}^*$ outputs $\mathcal{A}$'s output as it is, for every $s \in [q]$ and $i \in [\ell - 1]$, we have

$$\begin{aligned}&\Pr_{D_{\mathsf{real}}}[\mathcal{B}^*(\mathsf{pp}', k, \mathsf{pk}, \mathsf{sk}, \mathsf{H}_k(\mathsf{c}_{\mathsf{KEM}})) = 1 \mid t = s \wedge j = i] \\ &= \Pr_{D_{\mathsf{rand}}}[\mathcal{B}^*(\mathsf{pp}', k, \mathsf{pk}, \mathsf{sk}, y) = 1 \mid t = s \wedge j = i + 1].\end{aligned} \tag{7}$$

- From $\mathcal{A}$'s viewpoint, for every $s \in [q - 1]$, the experiments simulated by $\mathcal{B}^*$ in each of the following situations are identical:
  - $\mathcal{B}^*$'s input is an output of H, $t = s$, and $j = \ell$; and
  - $\mathcal{B}^*$'s input is a random value, $t = s + 1$, and $j = 1$.

  Since $\mathcal{B}^*$ outputs $\mathcal{A}$'s output as it is, for every $s \in [q - 1]$, we have

$$\begin{aligned}&\Pr_{D_{\mathsf{real}}}[\mathcal{B}^*(\mathsf{pp}', k, \mathsf{pk}, \mathsf{sk}, \mathsf{H}_k(\mathsf{c}_{\mathsf{KEM}})) = 1 \mid t = s \wedge j = \ell] \\ &= \Pr_{D_{\mathsf{rand}}}[\mathcal{B}^*(\mathsf{pp}', k, \mathsf{pk}, \mathsf{sk}, y) = 1 \mid t = s + 1 \wedge j = 1].\end{aligned} \tag{8}$$

Hence, we can compute $\mathcal{B}^*$'s distinguishing advantage as follows:

$$
\begin{aligned}
&\Big|\Pr_{D_{\mathsf{real}}}[\mathcal{B}^*(\mathsf{pp}', k, \mathsf{pk}, \mathsf{sk}, \mathsf{H}_k(\mathsf{c}_{\mathsf{KEM}})) = 1] - \Pr_{D_{\mathsf{rand}}}[\mathcal{B}^*(\mathsf{pp}', k, \mathsf{pk}, \mathsf{sk}, y) = 1]\Big| \\
&= \Big|\sum_{s\in[q]}\sum_{i\in[\ell]} \Pr_{t,j}[t = s \wedge j = i] \\
&\qquad \Big(\Pr_{D_{\mathsf{real}}}[\mathcal{B}^*(\mathsf{pp}', k, \mathsf{pk}, \mathsf{sk}, \mathsf{H}_k(\mathsf{c}_{\mathsf{KEM}})) = 1 \mid t = s \wedge j = i] \\
&\qquad\qquad - \Pr_{D_{\mathsf{rand}}}[\mathcal{B}^*(\mathsf{pp}', k, \mathsf{pk}, \mathsf{sk}, y) = 1 \mid t = s \wedge j = i]\Big)\Big| \\
&= \Big|\sum_{s\in[q]} \Pr_{t}[t = s] \sum_{i\in[\ell]} \Pr_{j}[j = i] \\
&\qquad \Big(\Pr_{D_{\mathsf{real}}}[\mathcal{B}^*(\mathsf{pp}', k, \mathsf{pk}, \mathsf{sk}, \mathsf{H}_k(\mathsf{c}_{\mathsf{KEM}})) = 1 \mid t = s \wedge j = i] \\
&\qquad\qquad - \Pr_{D_{\mathsf{rand}}}[\mathcal{B}^*_{D_{\mathsf{rand}}}(\mathsf{pp}', k, \mathsf{pk}, \mathsf{sk}, y) = 1 \mid t = s \wedge j = i]\Big)\Big| \\
&= \frac{1}{q\ell}\Big|\Pr_{D_{\mathsf{real}}}[\mathcal{B}^*(\mathsf{pp}', k, \mathsf{pk}, \mathsf{sk}, \mathsf{H}_k(\mathsf{c}_{\mathsf{KEM}})) = 1 = 1 \mid t = q \wedge j = \ell] \\
&\qquad\qquad - \Pr_{D_{\mathsf{rand}}}[\mathcal{B}^*(\mathsf{pp}', k, \mathsf{pk}, \mathsf{sk}, y) = 1 \mid t = 1 \wedge j = 1]\Big| \qquad (9) \\
&= \frac{1}{q\ell}\Big|\Pr[\mathsf{G}_4] - \Pr[\mathsf{G}_3]\Big|, \qquad (10)
\end{aligned}
$$

where $k \in \mathcal{K}, y \in \{0,1\}, t \in [q]$, and $j \in [\ell]$ are chosen uniformly at random, and $(\mathsf{c}_{\mathsf{KEM}}, \mathsf{K}) \xleftarrow{\$} \mathsf{Encap}(\mathsf{pk})$. Eq. (9) follows from Eqs. (7) and (8), and Eq. (10) follows from Eqs. (5) and (6). By Lemma 12, there exist PPT adversaries $\mathcal{B}''_0$ and $\mathcal{B}''_1$ such that

$$
\begin{aligned}
&\Big|\Pr_{D_{\mathsf{real}}}[\mathcal{B}^*(\mathsf{pp}', k, \mathsf{pk}, \mathsf{sk}, \mathsf{H}_k(\mathsf{c}_{\mathsf{KEM}})) = 1] - \Pr_{D_{\mathsf{rand}}}[\mathcal{B}^*(\mathsf{pp}', k, \mathsf{pk}, \mathsf{sk}, y) = 1]\Big| \\
&\le \mathbf{SD}(D_{\mathsf{real}}, D_{\mathsf{rand}}) \\
&\le \frac{1}{\sqrt{2}}\left(\mathsf{Adv}^{\mathsf{PRF}}_{\mathsf{H},\mathcal{B}''_0}(\lambda) + \mathsf{Adv}^{\mathsf{PR}}_{\mathsf{KEM},\mathcal{B}''_1}(\lambda) + 2^{-\ell_{\mathsf{KEM}}}\right)^{\frac{1}{2}}.
\end{aligned}
$$

Putting it all together, we have

$$
\Big|\Pr[\mathsf{G}_3] - \Pr[\mathsf{G}_4]\Big| \le \frac{q\ell}{\sqrt{2}}\left(\mathsf{Adv}^{\mathsf{PRF}}_{\mathsf{H},\mathcal{B}''_0}(\lambda) + \mathsf{Adv}^{\mathsf{PR}}_{\mathsf{KEM},\mathcal{B}''_1}(\lambda) + 2^{-\ell_{\mathsf{KEM}}}\right)^{\frac{1}{2}}.
$$

□ (**Lemma 16**)

Lemma 17. *There exist PPT adversaries $\mathcal{B}'''_0$ and $\mathcal{B}'''_1$ such that*

$$
\begin{aligned}
&\Big|\Pr[\mathsf{G}_4] - \Pr[\mathsf{G}_5]\Big| \\
&\le q\ell\Big((\lambda+1)\mathsf{Adv}^{\mathtt{CTX\text{-}PR}}_{\mathsf{KEM},\mathcal{B}'''_0}(\lambda) + \mathsf{Adv}^{\mathsf{PRF}}_{\mathsf{H},\mathcal{B}'''_1}(\lambda) + \lambda(\lambda+1)\cdot 2^{-(\ell+1)} + 2^{-\lambda}\Big).
\end{aligned}
$$

Proof. For $s \in [q]$, let $\mathsf{Bad}_{4,s}$ be the event in Game 4 that, in the process of generating the response to the s-th query during the game, the "do-until" loop for some $i \in [\ell]$ terminates because the counter $\mathtt{ctr}$ reaches λ before the condition $\mathtt{b}^* = \mathtt{b}_i$ is satisfied. Formally, let $\mathsf{Bad}_{4,s}$ be the event that the condition "$\mathtt{b}^* \neq \mathtt{b}_i \ \wedge \ \mathtt{ctr} \geq \lambda$" holds in some i-th itelation of "for" loop for s-th query. Even in Game 5, $\mathsf{Bad}_{5,s}$ can be defined in the same manner by conceptually maintaining $\mathtt{ctr}$ without using it for any conditional evaluation. Hence, it is easy to see that

$$\Pr[\mathsf{Bad}_{4,s}] = \Pr[\mathsf{Bad}_{5,s}]$$

holds. $\mathsf{Bad}_{4,s}$ and $\mathsf{Bad}_{5,s}$ can occur with the same probability, and if the bad event does not occur, the two games, Game 4 and Game 5, behave identically. Therefore, it holds

$$\Pr\Big[\mathsf{G}_4 \wedge \bigwedge_{s\in[q]} \overline{\mathsf{Bad}_{4,s}}\Big] = \Pr\Big[\mathsf{G}_5 \wedge \bigwedge_{s\in[q]} \overline{\mathsf{Bad}_{4,s}}\Big].$$

Thus, we have

$$\begin{aligned}\Big|\Pr[\mathsf{G}_4] - \Pr[\mathsf{G}_5]\Big| &= \Bigg|\Big(\Pr\Big[\mathsf{G}_4 \wedge \bigvee_{s\in[q]} \mathsf{Bad}_{4,s}\Big] + \Pr\Big[\mathsf{G}_4 \wedge \bigwedge_{s\in[q]} \overline{\mathsf{Bad}_{4,s}}\Big]\Big) \\ &\qquad - \Big(\Pr\Big[\mathsf{G}_5 \wedge \bigvee_{s\in[q]} \mathsf{Bad}_{5,s}\Big] + \Pr\Big[\mathsf{G}_5 \wedge \bigwedge_{s\in[q]} \overline{\mathsf{Bad}_{5,s}}\Big]\Big)\Bigg| \\ &\leq \sum_{s\in[q]} \Pr[\mathsf{Bad}_{4,s}].\end{aligned}$$

Next, we will show that there exist PPT adversaries $\mathcal{B}_0'''$ and $\mathcal{B}_1'''$ such that

$$\sum_{s\in[q]} \Pr[\mathsf{Bad}_{4,s}] \leq q\ell\Big((\lambda{+}1)\mathsf{Adv}^{\mathtt{CTX\text{-}PR}}_{\mathsf{KEM},\mathcal{B}_0'''}(\lambda){+}\mathsf{Adv}^{\mathtt{PRF}}_{\mathsf{H},\mathcal{B}_1'''}(\lambda){+}\lambda(\lambda{+}1)\cdot 2^{-(\ell+1)}{+}2^{-\lambda}\Big).$$

For $s \in [q]$ and $i \in [\ell]$, let $\mathsf{Bad}_{4,s,i}$ be the specific event in Game 4 that the condition "$\mathtt{b}^* \neq \mathtt{b}_i \wedge \mathtt{ctr} \geq \lambda$" holds in the i-th iteration of the "for" loop for the s-th query. By the definition of $\mathsf{Bad}_{4,s,i}$, we have

$$\begin{aligned}\Pr[\mathsf{Bad}_{4,s}] &= \Pr[\mathsf{Bad}_{4,s,1} \vee \mathsf{Bad}_{4,s,2} \vee \cdots \vee \mathsf{Bad}_{4,s,\ell}] \\ &\leq \sum_{i\in[\ell]} \Pr[\mathsf{Bad}_{4,s,i}].\end{aligned}$$

The situation that $\mathtt{b}^* \neq \mathtt{b}_i \wedge \mathtt{ctr} \geq \lambda$ occurs in the execution inside the "do-until" loop of fRand is exactly the same as the event for ϵ in Lemma 11. Hence, for every $s \in [q]$ and $t \in [\ell]$, $\Pr[\mathsf{Bad}_{4,s,i}]$ is bounded by ϵ. Therefore, from the above,

Lemma 11, and the union bound, we can say that there exist PPT adversaries $\mathcal{B}_0'''$ and $\mathcal{B}_1'''$ such that

$$\begin{aligned}
&\sum_{s\in[q]} \Pr[\mathsf{Bad}_{4,s}] \\
&\leq \sum_{s\in[q]}\sum_{i\in[\ell]} \Pr[\mathsf{Bad}_{4,s,i}] \\
&\leq q\ell\Big((\lambda+1)\mathsf{Adv}^{\mathtt{CTX\text{-}PR}}_{\mathsf{KEM},\mathcal{B}_0'''}(\lambda) + \mathsf{Adv}^{\mathtt{PRF}}_{\mathsf{H},\mathcal{B}_1'''}(\lambda) + \lambda(\lambda+1)\cdot 2^{-(\ell+1)} + 2^{-\lambda}\Big),
\end{aligned}$$

which completes the proof. □ (**Lemma 17**)

Lemma 18. $\Pr[\mathsf{G}_5] = \Pr[\mathsf{G}_6]$ *holds.*

Proof. Fix $\mathcal{A}$'s query, $i \in [n]$, and b_i, and let us focus on the distribution of r_i. Let P be the predicate defined by $\mathsf{P}(r) := (\mathsf{H}_k(\mathsf{Encap}(\mathsf{fpk}_i; r)) \stackrel{?}{=} \mathsf{b}_i)$. Then, it is straightforward to see that the set $\mathcal{R}_i$ defined in Game 6 is exactly $\mathsf{P}^{-1}(1)$, from which r_i is chosen uniformly at random in Game 6. On the other hand, in Game 5, the distribution of r_i can be represented as follows:

$$\Big\{h \leftarrow 0;\ \text{do } \{h \leftarrow h+1; r_h \stackrel{\$}{\leftarrow} \mathcal{R}\} \text{ until } \mathsf{P}(r_h) = 1 : r_h\Big\}.$$

Hence, invoking Lemma 1 with the above predicate P, we can see that the distribution of r_i in Game 6 is identical to the distribution of r_i in Game 5, which implies $\Pr[\mathsf{G}_5] = \Pr[\mathsf{G}_6]$ as desired.

□ (**Lemma 18**)

Lemma 19. $\Pr[\mathsf{G}_6] = \Pr[\mathsf{G}_7]$ *holds.*

Proof. In Game 7, for every $i \in [\ell]$, the distribution of r_i is uniform over $\mathcal{R}$.

Let f be the function defined by $f(r) := \mathsf{H}_k(\mathsf{Encap}(\mathsf{fpk}_i; r))$. In Game 6, for every $i \in [\ell]$, the distribution of r_i is denoted as follows:

$$\begin{aligned}
&\left\{\begin{array}{l}\widehat{r} \stackrel{\$}{\leftarrow} \mathcal{R};\\ \mathsf{b}_i \leftarrow \mathsf{H}_k(\mathsf{Encap}(\mathsf{fpk}_i; \widehat{r}));\\ r_i \stackrel{\$}{\leftarrow} \Big\{r \in \mathcal{R} \;\Big|\; \mathsf{H}_k(\mathsf{Encap}(\mathsf{fpk}_i; r)) = \mathsf{b}_i\Big\}\end{array} : r_i\right\} \\
&\equiv \Big\{\widehat{r} \stackrel{\$}{\leftarrow} \mathcal{R}; \mathsf{b}_i \leftarrow f(\widehat{r}); r_i \stackrel{\$}{\leftarrow} f^{-1}(\mathsf{b}_i) : r_i\Big\}
\end{aligned}$$

For every $i \in [\ell]$, by Lemma 2 with $f(r) := \mathsf{H}_k(\mathsf{Encap}(\mathsf{fpk}_i; r))$,

$$\Big\{r \stackrel{\$}{\leftarrow} \mathcal{R} : r\Big\} \equiv \Big\{\widehat{r} \stackrel{\$}{\leftarrow} \mathcal{R}; \mathsf{b}_i \leftarrow f(\widehat{r}); r_i \stackrel{\$}{\leftarrow} f^{-1}(\mathsf{b}_i) : r_i\Big\}$$

holds. Therefore, we can see that r_i has the same distribution in Games 6 and 7, and thus we have $\Pr[\mathsf{G}_6] = \Pr[\mathsf{G}_7]$.

□ (**Lemma 19**)

□ (**Theorem 10**)

6 Wang et al.'s Second SAME Scheme

In this section, we investigate whether Wang et al.'s second scheme [29, Section 5] satisfies our refined security notions. Wang et al.'s second scheme is an $(\ell+1)$-SAME scheme where ℓ denotes the length of duplicate messages one wishes to send. Their $(\ell+1)$-SAME scheme is constructed for hybrid encryption, consisting of a so-called "module-level" KEM scheme and a DEM scheme. A module-level KEM is a type of KEMs that satisfy some structural properties that are satisfied by many of existing KEM (the relatives of the ElGamal KEM).

Our results on Wang et al.'s second scheme are as follows: We show that their second scheme satisfies `SAME-R` security by assuming a special property that (a part of) the session-key generation algorithm of the underlying KEM used as a building block behaves like a PRF when the public key is not given to an adversary. (This property is called the *special PRF property*.) On the other hand, we also show that their scheme does not satisfy `S-SAME-R` security as there is an easy attack.

Definition of a Module-Level KEM. We first recall the definition of a module-level KEM whose definition is taken from [29, Section 5]. A module-level KEM scheme KEM consists of the four PPT algorithms $(\mathsf{KSetup}, \mathsf{KKG}, \mathsf{Encap}, \mathsf{Decap})$ which have the following structural properties.

$\mathsf{KSetup}(1^\lambda) \xrightarrow{\$} \mathsf{pp}$**:** This is the setup algorithm that takes a security parameter 1^λ as input, and outputs a public parameter pp.

$\mathsf{KKG}(\mathsf{pp}) \xrightarrow{\$} (\mathsf{pk}, \mathsf{sk})$**:** This is the key generation algorithm that takes a public parameter pp as input, and outputs a pair of public key $\mathsf{pk} = (\mathsf{ek}, \mathsf{tk})$ and secret key $\mathsf{sk} = (\mathsf{dk}, \mathsf{vk})$ by running the following subalgorithms.

- $\mathsf{KEMEk}(\mathsf{pp}) \xrightarrow{\$} (\mathsf{ek}, \mathsf{dk})$: It outputs a pair of encapsulation key ek and decapsulation key dk.
- $\mathsf{KEMTk}(\mathsf{pp}) \xrightarrow{\$} (\mathsf{tk}, \mathsf{vk})$: It outputs a pair of tag generation key tk and verification key vk.

$\mathsf{Encap}(\mathsf{pk}) \xrightarrow{\$} (\psi, \mathsf{K})$**:** This is the encapsulation algorithm that takes a public key pk, and outputs a ciphertext $\psi = (\mathsf{c}, \pi)$ and a session-key K by sampling a randomness $r \in \mathcal{R}_{\mathsf{KEM}}$ uniformly at random[7] and running the following subalgorithms.

- $\mathsf{KEMKg}(\mathsf{ek}, r) \to \mathsf{K}$: It outputs a session-key K.
- $\mathsf{KEMCg}(r) \to \mathsf{c}$: It outputs a ciphertext c.
- $\mathsf{KEMTg}(\mathsf{tk}, r) \to \pi$: It outputs a ciphertext tag π of c.

$\mathsf{Decap}(\mathsf{sk}, \psi) \to \mathsf{K}'/\perp$**:** This is the decapsulation algorithm that takes a secret key sk and a ciphertext $\psi = (\mathsf{c}, \pi)$. It is run by the following subalgorithms.

- $\mathsf{KEMKd}(\mathsf{dk}, \mathsf{c}) \to \mathsf{K}'$: It outputs a session-key K'.

[7] Wang et al. [29] defined another algorithm KEMRg for the randomness sampling as a subalgorithm of Encap. For simplicity and avoiding redundancy, we do not introduce KEMRg and sample the randomness directly instead.

– $\mathsf{KEMVf}(\mathsf{vk}, \mathsf{c}) \to \pi'$: It outputs a ciphertext tag π' of c.

If $\pi' = \pi$, then Decap outputs K'. Otherwise, it outputs $\perp$ to indicate a decryption failure.

As explained in [29], many existing KEM schemes can be captured as a module-level KEM.

We will use a module-level KEM scheme with the following property.

Definition 12. *Let* KEM *be a module-level KEM scheme. We say that* KEM *satisfies universal decryptability if the following probability is negligible:*

$$\Pr\left[\begin{array}{l} \mathsf{pp}' \xleftarrow{\$} \mathsf{KSetup}(1^\lambda); \\ (\mathsf{ek}, \mathsf{dk}) \xleftarrow{\$} \mathsf{KEMEk}(\mathsf{pp}'); \\ r \xleftarrow{\$} \mathcal{R}_{\mathsf{KEM}}; \\ \mathsf{c} \leftarrow \mathsf{KEMCg}(r) \end{array} : \mathsf{KEMKg}(\mathsf{ek}, r) \neq \mathsf{KEMKd}(\mathsf{dk}, \mathsf{c}) \right].$$

Formal Description of Wang et al.'s Second Scheme and Its Security. Let $\mathsf{KEM} = (\mathsf{KSetup}, \mathsf{KKG}, \mathsf{Encap}, \mathsf{Decap})$ be a module-level KEM scheme, and thus it has the subalgorithms $(\mathsf{KEMEk}, \mathsf{KEMTk}, \mathsf{KEMKg}, \mathsf{KEMCg}, \mathsf{KEMTg}, \mathsf{KEMKd}, \mathsf{KEMVf})$. Let the KEM-ciphertext length be $\ell_{\mathsf{KEM}} = \ell_{\mathsf{KEM}}(\lambda)$ and the session-key length be λ. Furthermore, we require that the randomness space of Encap, $\mathcal{R}_{\mathsf{KEM}}$, be an additive and cyclic group with a generator $1_{\mathcal{R}_{\mathsf{KEM}}}$. Let $\mathsf{DEM} = (\mathsf{DEnc}, \mathsf{DDec})$ be a DEM scheme. Let $\mathsf{H} : \mathcal{K} \times \{0,1\}^\lambda \to \mathcal{R}_{\mathsf{KEM}}$ be an efficiently computable function, for which we write $\mathsf{H}_k(\cdot)$ to mean $\mathsf{H}(k, \cdot)$ for simplicity. Then, Wang et al.'s second $(\ell+1)$-SAME scheme $\mathsf{AME}_{\mathtt{WCHY2}} = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{fRand}, \mathsf{dDec})$ is constructed as in Fig. 14.

Wang et al. showed the following theorem that guarantees the correctness of $\mathsf{AME}_{\mathtt{WCHY2}}$.

Theorem 11. *If* KEM *satisfies correctness and universal decryptability, then* $\mathsf{AME}_{\mathtt{WCHY2}}$ *satisfies correctness.*

Regarding the the $\mathtt{S\text{-}SAME}$ security of $\mathsf{AME}_{\mathtt{WCHY2}}$, Wang et al. showed the following theorem.

Theorem 12. *If* KEM *satisfies universal decryptability, key pseudorandomness, and the homomorphic property*[8]*,* DEM *satisfies* $\mathtt{IND\text{-}OT}$ *security, and* H *is entropy-smoothing, then* $\mathsf{AME}_{\mathtt{WCHY2}}$ *satisfies* $\mathtt{S\text{-}SAME}$ *security.*

Here, the homomorphic property refers to the property that both KEMKg and KEMCg are homomorphic with respect to their randomness inputs, and the key pseudorandomness refers to the indistinguishability between a session-key K generated by $\mathsf{KEMKg}(\mathsf{ek}, r)$ and a key K sampled uniformly at random from $\mathcal{K}$ where $\mathsf{pp} \xleftarrow{\$} \mathsf{KSetup}(1^\lambda)$, $(\mathsf{ek}, \mathsf{dk}) \xleftarrow{\$} \mathsf{KEMEk}(\mathsf{pp})$, and $r \xleftarrow{\$} \mathcal{R}_{\mathsf{KEM}}$.[9]

[8] For the formal definition, see [29].

[9] We omit the formal definitions of key pseudorandomness and the homomorphic property since we do not use them directly for our results. Note that key pseudorandomness of a module-level KEM is implied by its $\mathtt{IND\text{-}CPA}$ security. For the detailed definitions, see [29].

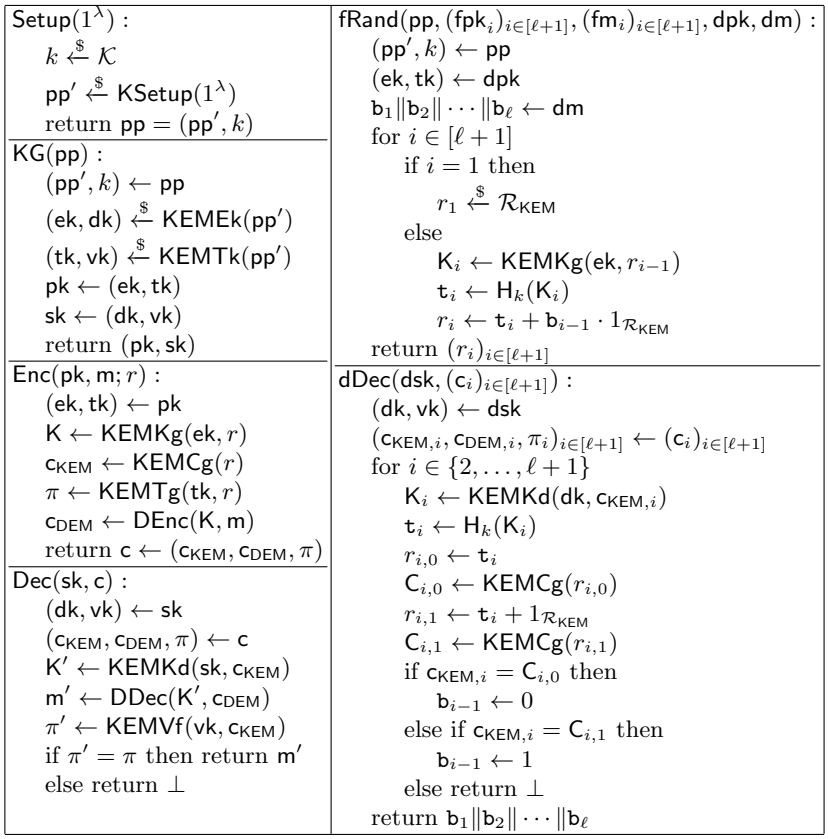

Fig. 14. Wang et al.'s second $(\ell + 1)$-SAME scheme $\mathsf{AME}_{\mathtt{WCHY2}}$ constructed from a module-level KEM scheme which satisfies universal decryptability, key pseudorandomness, and the homomorphic property, an IND-OT secure DEM scheme, and an entropy-smoothing function. In the figure, $\mathtt{b}_i \in \{0, 1\}$ for every $i \in [\ell]$.

As the main result in this section, we show that $\mathsf{AME}_{\mathtt{WCHY2}}$ satisfies SAME-R security. For showing this, we need to rely on a special property of the underlying module-level KEM that its KEMKg algorithm has a property similar to a PRF with its key ek. Thus, we introduce its formal definition below.

Definition 13. *Let* KEM *be a module-level KEM scheme whose session-key space is* $\{0, 1\}^\lambda$*. We say that* KEM *satisfies the* special PRF property *if for any PPT adversary* $\mathcal{A}$*, the following advantage is negligible:*

$$\mathsf{Adv}^{\mathsf{PRF}}_{\mathsf{KEM},\mathcal{A}}(\lambda) := \left| \Pr[\mathcal{A}^{\mathsf{KEMKg}(\mathsf{ek},\cdot)}(\mathsf{pp}) = 1] - \Pr\left[\mathcal{A}^{f(\cdot)}(\mathsf{pp}) = 1\right] \right|,$$

where $\mathsf{pp} \xleftarrow{\$} \mathsf{KSetup}(1^\lambda)$*,* $(\mathsf{ek}, \mathsf{dk}) \xleftarrow{\$} \mathsf{KEMEk}(\mathsf{pp})$*, and* $f \xleftarrow{\$} \mathsf{FUNC}_{\mathcal{R}_{\mathsf{KEM}} \to \{0,1\}^\ell}$*.*

We stress that although ek is a part of a public key of a module-level KEM, it is treated analogously to a (secret) PRF key, i.e., it is not given to the adversary in the security experiment.

One might wonder whether this property is satisfied by existing module-level KEMs. We observe that the hashed ElGamal KEM (also called DHIES [2]) in which the session-key component in the original (non-hashed) ElGamal KEM is hashed by a cryptographic hash function, satisfies this property in the random oracle model. Since many of the relatives of the ElGamal KEM, e.g. [7, 13, 18, 20, 22, 23], are captured as a module-level KEM and have the ElGamal KEM as their internal structure, these KEMs also satisfy the special PRF property when (the session-key component is hashed, and) the used hash function is modeled as a random oracle. For completeness, we formally show that the hashed ElGamal KEM has the special PRF property in Appendix E.

We now state our results on Wang et al.'s second SAME scheme. The following theorem guarantees the $\mathtt{SAME\text{-}R}$ security of $\mathsf{AME}_{\mathtt{WCHY2}}$.

Theorem 13. *If* KEM *satisfies pseudorandomness and the special PRF property,* DEM *satisfies* $\mathtt{IND\text{-}OT}$ *security, and* H *is entropy-smoothing, then* $\mathsf{AME}_{\mathtt{WCHY2}}$ *satisfies* $\mathtt{SAME\text{-}R}$ *security.*

Proof. Since $\mathtt{IND\text{-}CPA}$ security of hybrid encryption follows from the pseudorandomness of KEM and $\mathtt{IND\text{-}OT}$ security of DEM, $(\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ is $\mathtt{IND\text{-}CPA}$ secure.

By Theorem 2, it is sufficient to show that $\mathsf{AME}_{\mathtt{WCHY2}}$ is $\mathtt{SAME\text{-}R'}$ secure. Let $\mathcal{A}$ be a PPT adversary against the $\mathtt{SAME\text{-}R'}$ security that makes q queries to the randomness oracle. We will show that for $\mathcal{A}$, there exist PPT adversaries $\mathcal{B}$ and $\mathcal{B}'$ that satisfy

$$\mathsf{Adv}^{\mathtt{SAME\text{-}R'}}_{\mathsf{AME}_{\mathtt{WCHY2}},\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathtt{PRF}}_{\mathsf{KEM},\mathcal{B}}(\lambda) + q\ell\mathsf{Adv}^{\mathtt{ESF}}_{\mathsf{H},\mathcal{B}'}(\lambda) + \frac{q\ell(\ell-1)}{2|\mathcal{R}_{\mathsf{KEM}}|}. \tag{11}$$

We show the above by using a sequence of games: Game 0, Game 1, and Game 2 for $\mathcal{A}$.

Game 0: This is exactly $\mathsf{Expt}^{\mathtt{SAME\text{-}R'}}_{\mathsf{AME}_{\mathtt{WCHY2}},\mathcal{A}}(\lambda)$ in which its challenge bit $b = 0$.

Game 1: This game is the same as Game 0 except for the initialization procedure and how session-keys $\mathsf{K}_2, \ldots, \mathsf{K}_{\ell+1} \in \{0,1\}^\lambda$ are generated in the randomness oracle. Specifically, $f \in \mathsf{FUNC}_{\mathcal{R}_{\mathsf{KEM}} \to \{0,1\}^\lambda}$ is chosen uniformly at random at the beginning of the game, and for each randomness oracle query, the session-keys are computed as $\mathsf{K}_{i+1} \leftarrow f(r_i)$ for $i \in [\ell]$.

Game 2: This game is the same as Game 1 except for how $r_2, \ldots, r_{\ell+1}$ are generated. Specifically, for each $i \in [\ell+1]$, $r_i \in \mathcal{R}_{\mathsf{KEM}}$ is chosen uniformly at random. (By definition, Game 2 is exactly $\mathsf{Expt}^{\mathtt{SAME\text{-}R'}}_{\mathsf{AME},\mathcal{A}}(\lambda)$ where its challenge bit is $b = 1$.)

Let G_N be the event that $\mathcal{A}$ returns 1 in Game N. Then, by the definition of the events and the triangle inequality, we have,

$$\mathsf{Adv}^{\mathtt{SAME\text{-}R'}}_{\mathsf{AME},\mathcal{A}}(\lambda) = \Big|\Pr[\mathsf{G}_0] - \Pr[\mathsf{G}_2]\Big| \leq \Big|\Pr[\mathsf{G}_0] - \Pr[\mathsf{G}_1]\Big| + \Big|\Pr[\mathsf{G}_1] - \Pr[\mathsf{G}_2]\Big|. \quad (12)$$

Due to Eq. (12) and the following Lemmas 20 and 21, we can conclude that there exist PPT adversaries $\mathcal{B}$ and $\mathcal{B}'$ that satisfy Eq. (11).

Lemma 20. *There exists a PPT adversary $\mathcal{B}$ such that*

$$\Big|\Pr[\mathsf{G}_0] - \Pr[\mathsf{G}_1]\Big| = \mathsf{Adv}^{\mathtt{PRF}}_{\mathsf{KEM},\mathcal{B}}(\lambda).$$

Proof. We construct an adversary $\mathcal{B}$ against the special PRF property of KEM as follows:

$\mathcal{B}^{f(\cdot)}(\mathsf{pp}')$**:** $\mathcal{B}$ picks $k \in \mathcal{K}$ uniformly at random, sets $\mathsf{pp} \leftarrow (\mathsf{pp}', k)$, and computes $(\mathsf{fpk}_i, \mathsf{fsk}_i) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp})$ for each $i \in [\ell+1]$. Then, $\mathcal{B}$ runs $\mathcal{A}(\mathsf{pp}, (\mathsf{fpk}_i)_{i\in[\ell+1]})$. For each query $((\mathsf{fm}_i)_{i\in[\ell+1]}, \mathsf{dm})$ from $\mathcal{A}$, $\mathcal{B}$ executes the following procedure:
$\mathcal{B}$ parses $\mathsf{b}_1\|\mathsf{b}_2\|\cdots\|\mathsf{b}_\ell \leftarrow \mathsf{dm}$, and picks $r_1 \in \mathcal{R}_{\mathsf{KEM}}$ uniformly at random. For generating r_i for each $i \in \{2,\ldots,\ell+1\}$, $\mathcal{B}$ submits r_{i-1} to the function oracle, receives its result K_i, and computes $\mathsf{t}_i \leftarrow \mathsf{H}_k(\mathsf{K}_i)$ and $r_i \leftarrow \mathsf{t}_i + \mathsf{b}_{i-1} \cdot 1_{\mathcal{R}_{\mathsf{KEM}}}$. Finally $\mathcal{B}$ returns $(r_i)_{i\in[\ell+1]}$ to $\mathcal{A}$.
Finally, $\mathcal{A}$ returns a bit b' and $\mathcal{B}$ returns b' as it is.

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

- If $\mathcal{B}$ is accessing the real function $\mathsf{KKG}(\mathsf{ek},\cdot)$, then $\mathcal{B}$ simulates Game 0 perfectly for $\mathcal{A}$, where the encapsulation key ek generated by $\mathcal{B}$'s experiment is implicitly regarded as the first component of dpk in the experiment for $\mathcal{A}$. Since $\mathcal{B}$ outputs $\mathcal{A}$'s output as it is, we have
$$\Pr_{(\mathsf{ek},\mathsf{tk})\xleftarrow{\$}\mathsf{KEMEk}(\mathsf{pp}')}[\mathcal{B}^{\mathsf{KEMKg}(\mathsf{ek},\cdot)}(\mathsf{pp}') = 1] = \Pr[\mathsf{G}_0].$$
- If $\mathcal{B}$ is accessing a random function, then $\mathcal{B}$ simulates Game 1 perfectly for $\mathcal{A}$. Since $\mathcal{B}$ outputs $\mathcal{A}$'s output as it is, we have
$$\Pr_{f\xleftarrow{\$}\mathsf{FUNC}_{\mathcal{R}_{\mathsf{KEM}}\to\{0,1\}^\lambda}}\big[\mathcal{B}^{f(\cdot)}(\mathsf{pp}') = 1\big] = \Pr[\mathsf{G}_1].$$

Hence, we have

$$\begin{aligned}
&\mathsf{Adv}^{\mathtt{PRF}}_{\mathsf{KEM},\mathcal{B}}(\lambda)\\
&= \left|\Pr_{(\mathsf{ek},\mathsf{tk})\xleftarrow{\$}\mathsf{KEMEk}(\mathsf{pp}')}[\mathcal{B}^{\mathsf{KEMKg}(\mathsf{ek},\cdot)}(\mathsf{pp}') = 1] - \Pr_{f\xleftarrow{\$}\mathsf{FUNC}_{\mathcal{R}_{\mathsf{KEM}}\to\{0,1\}^\lambda}}\big[\mathcal{B}^{f(\cdot)}(\mathsf{pp}') = 1\big]\right|\\
&= \Big|\Pr[\mathsf{G}_0] - \Pr[\mathsf{G}_1]\Big|.
\end{aligned}$$

This completes the proof. □ (**Lemma 20**)

Lemma 21. *There exists a PPT adversary $\mathcal{B}'$ such that*

$$\left|\Pr[\mathsf{G}_1] - \Pr[\mathsf{G}_2]\right| \leq q\ell \mathsf{Adv}^{\mathtt{ESF}}_{\mathsf{H},\mathcal{B}'}(\lambda) + \frac{q\ell(q\ell-1)}{2|\mathcal{R}_{\mathsf{KEM}}|}.$$

Proof. For showing the lemma, we introduce the following $q+1$ intermediate games between Game 1 and Game 2.

Game $1.s$ *(for* $s \in \{0,1,\ldots,q\}$*):* This game is the same as Game 1 except for the procedure of the randomness oracle. Specifically, it behaves as in Game 1 for oracle accesses up to the $(q-s)$-th query, and as in Game 2 for subsequent oracle accesses.

By definition, Game 1 is identical to Game 1.0, and Game 2 is identical to Game $1.q$, namely, it holds that

$$\Pr[\mathsf{G}_{1.0}] = \Pr[\mathsf{G}_1], \tag{13}$$

and

$$\Pr[\mathsf{G}_{1.q}] = \Pr[\mathsf{G}_2]. \tag{14}$$

In words, we start from Game 1 (i.e. Game 1.1) and, beginning with the last (the q-th) query, gradually modify the oracle responses one by one, moving through hybrid games until we reach Game 2 (i.e. Game $1.q$).

Furthermore, we introduce the following additional intermediate games between Game $1.s$ and Game $1.(s+1)$ for each $s \in \{0,1,\ldots,q-1\}$.

Game $1.s.i$ *(where* $s \in \{0,1,\ldots,q-1\}$ *and* $i \in [\ell+1]$*):* This game is the same as Game $1.s$ except for how the response of the randomness oracle for the $(q-s)$-th query is generated. Specifically, for the $(q-s)$-th query, $r_e \in \mathcal{R}_{\mathsf{KEM}}$ is chosen uniformly at random for each $e \in [i]$ (and, for each $e \in \{i+1,\ldots,\ell+1\}$, r_e is generated in the same manner as in Game 1).

Game $1.s.i.A$ *(where* $s \in \{0,1,\ldots,q-1\}$ *and* $i \in [\ell]$*):* This game is the same as Game $1.s.i$ except for how K_{i+1} for the response of the $(q-s)$-th query is generated. Specifically, $\mathsf{K}_{i+1} \in \{0,1\}^\lambda$ is chosen uniformly at random.

Game $1.s.i.B$ *(where* $s \in \{0,1,\ldots,q-1\}$ *and* $i \in [\ell]$*):* This game is the same as Game $1.s.i.A$ except for how $\mathtt{t}_{i+1}$ for the response of the $(q-s)$-th query is generated. Specifically, $\mathtt{t}_{i+1} \in \mathcal{R}_{\mathsf{KEM}}$ is chosen uniformly at random.

For convenience, Fig. 15 describes the behaviors of the randomness oracle in each game where $\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game}N}$ describes the randomness oracle in Game $N \in \{1.s.i,\ 1.s.e.A,\ 1.s.e.B\}$ for $s \in \{0,1,\ldots,q-1\}$, $i \in [\ell+1]$, and $e \in [\ell]$. Note

$\underline{\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game1}.s.i}((\mathsf{fm}_e)_{e\in[\ell+1]},\mathsf{dm}):}$

$(\mathsf{pp}', k) \leftarrow \mathsf{pp}$
$\mathtt{b}_1 \| \mathtt{b}_2 \| \cdots \| \mathtt{b}_\ell \leftarrow \mathsf{dm}$
$r_1 \xleftarrow{\$} \mathcal{R}_{\mathsf{KEM}}$
$\vdots$
$r_i \xleftarrow{\$} \mathcal{R}_{\mathsf{KEM}}$
$\mathsf{K}_{i+1} \leftarrow f(r_i)$
$\mathtt{t}_{i+1} \leftarrow \mathsf{H}_k(\mathsf{K}_{i+1})$
$r_{i+1} \leftarrow \mathtt{t}_{j+1} + \mathtt{b}_i \cdot 1_{\mathcal{R}_{\mathsf{KEM}}}$
$\mathsf{K}_{i+2} \leftarrow f(r_{i+1})$
$\mathtt{t}_{i+2} \leftarrow \mathsf{H}_k(\mathsf{K}_{i+2})$
$r_{i+2} \leftarrow \mathtt{t}_{i+2} + \mathtt{b}_{i+1} \cdot 1_{\mathcal{R}_{\mathsf{KEM}}}$
$\vdots$
$\mathsf{K}_{\ell+1} \leftarrow f(r_\ell)$
$\mathtt{t}_{\ell+1} \leftarrow \mathsf{H}_k(\mathsf{K}_{\ell+1})$
$r_{\ell+1} \leftarrow \mathtt{t}_{\ell+1} + \mathtt{b}_\ell \cdot 1_{\mathcal{R}_{\mathsf{KEM}}}$
return $(r_i)_{i\in[\ell+1]}$

$\underline{\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game1}.s.i.A}((\mathsf{fm}_e)_{e\in[\ell+1]},\mathsf{dm}):}$

$(\mathsf{pp}', k) \leftarrow \mathsf{pp}$
$\mathtt{b}_1 \| \mathtt{b}_2 \| \cdots \| \mathtt{b}_\ell \leftarrow \mathsf{dm}$
$r_1 \xleftarrow{\$} \mathcal{R}_{\mathsf{KEM}}$
$\vdots$
$r_i \xleftarrow{\$} \mathcal{R}_{\mathsf{KEM}}$
$\mathsf{K}_{i+1} \xleftarrow{\$} \{0,1\}^\lambda$
$\mathtt{t}_{i+1} \leftarrow \mathsf{H}_k(\mathsf{K}_{i+1})$
$r_{i+1} \leftarrow \mathtt{t}_{j+1} + \mathtt{b}_i \cdot 1_{\mathcal{R}_{\mathsf{KEM}}}$
$\mathsf{K}_{i+2} \leftarrow f(r_{i+1})$
$\mathtt{t}_{i+2} \leftarrow \mathsf{H}_k(\mathsf{K}_{i+2})$
$r_{i+2} \leftarrow \mathtt{t}_{i+2} + \mathtt{b}_{i+1} \cdot 1_{\mathcal{R}_{\mathsf{KEM}}}$
$\vdots$
$\mathsf{K}_{\ell+1} \leftarrow f(r_\ell)$
$\mathtt{t}_{\ell+1} \leftarrow \mathsf{H}_k(\mathsf{K}_{\ell+1})$
$r_{\ell+1} \leftarrow \mathtt{t}_{\ell+1} + \mathtt{b}_\ell \cdot 1_{\mathcal{R}_{\mathsf{KEM}}}$
return $(r_e)_{e\in[\ell+1]}$

$\underline{\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game1}.s.i.B}((\mathsf{fm}_e)_{e\in[\ell+1]},\mathsf{dm}):}$

$(\mathsf{pp}', k) \leftarrow \mathsf{pp}$
$\mathtt{b}_1 \| \mathtt{b}_2 \| \cdots \| \mathtt{b}_\ell \leftarrow \mathsf{dm}$
$r_1 \xleftarrow{\$} \mathcal{R}_{\mathsf{KEM}}$
$\vdots$
$r_i \xleftarrow{\$} \mathcal{R}_{\mathsf{KEM}}$
$\mathtt{t}_{i+1} \xleftarrow{\$} \mathcal{R}_{\mathsf{KEM}}$
$r_{i+1} \leftarrow \mathtt{t}_{j+1} + \mathtt{b}_i \cdot 1_{\mathcal{R}_{\mathsf{KEM}}}$
$\mathsf{K}_{i+2} \leftarrow f(r_{i+1})$
$\mathtt{t}_{i+2} \leftarrow \mathsf{H}_k(\mathsf{K}_{i+2})$
$r_{i+2} \leftarrow \mathtt{t}_{i+2} + \mathtt{b}_{i+1} \cdot 1_{\mathcal{R}_{\mathsf{KEM}}}$
$\vdots$
$\mathsf{K}_{\ell+1} \leftarrow f(r_\ell)$
$\mathtt{t}_{\ell+1} \leftarrow \mathsf{H}(\mathsf{K}_{\ell+1})$
$r_{\ell+1} \leftarrow \mathtt{t}_{\ell+1} + \mathtt{b}_\ell \cdot 1_{\mathcal{R}_{\mathsf{KEM}}}$
return $(r_i)_{i\in[\ell+1]}$

Fig. 15. Behaviors of the randomness oracle to respond to the $(q-s)$-th query in Game $1.s.i$ (where $i \in [\ell+1]$), Game $1.s.i.A$, and Game $1.s.i.B$ (where $i \in [\ell]$) for $s \in \{0,1,\ldots,q-1\}$. Note that $f : \mathcal{R}_{\mathsf{KEM}} \to \{0,1\}^\lambda$ is a random function.

that we consider transitions of the games in the following order for every $s \in \{0,1,\ldots,q-1\}$:

$$
\begin{aligned}
& 1.s \\
& \to\ 1.s.1 \ \to\ 1.s.1.A \ \to\ 1.s.1.B \\
& \to\ 1.s.2 \ \to\ \cdots \\
& \to\ \cdots \\
& \to 1.s.\ell \ \to\ 1.s.\ell.A \ \to\ 1.s.\ell.B \\
& \to 1.s.(\ell+1) \\
\to & 1.(s+1).
\end{aligned}
$$

In words, for the response to the $(q-s)$-th query, we are replacing r_{i+1} with truly random values in forward order, starting from the first element.

Regarding the relations among the games, the following hold.

– By the definition of Game $1.s.i$, for each $s \in \{0,\ldots,q-1\}$, Game $1.s$ is identical to Game $1.s.1$. Therefore, for every $s \in \{0,\ldots,q-1\}$, we have

$$\Pr[\mathsf{G}_{1.s}] = \Pr[\mathsf{G}_{1.s.1}]. \tag{15}$$

– By the definition of Game $1.s.i$, in Game $1.s.(\ell{+}1)$ for each $s \in \{0, \ldots, q-1\}$, $r_1, r_2, \ldots, r_{\ell+1} \in \mathcal{R}_{\mathsf{KEM}}$ are chosen uniformly at random. Therefore, Game $1.s.(\ell+1)$ is identical to Game $1.(s+1)$. Hence, for every $s \in \{0, \ldots, q-1\}$, we have

$$\Pr[\mathsf{G}_{1.s.\ell+1}] = \Pr[\mathsf{G}_{1.s+1}]. \tag{16}$$

Furthermore, we will show the following three claims.

Claim 1. *For every $s \in \{0, \ldots, q-1\}$ and $i \in [\ell]$, it holds that*

$$\left|\Pr[\mathsf{G}_{1.s.i}] - \Pr[\mathsf{G}_{1.s.i.A}]\right| \leq \frac{\ell(q-s)-i}{\left|\mathcal{R}_{\mathsf{KEM}}\right|}.$$

Claim 2. *There exists a PPT adversary $\mathcal{B}'$ such that*

$$\left|\sum_{s=0}^{q-1}\sum_{i=2}^{\ell+1}\left(\Pr[\mathsf{G}_{1.s.i.A}] - \Pr[\mathsf{G}_{1.s.i.B}]\right)\right| = q\ell\mathsf{Adv}^{\mathtt{ESF}}_{\mathsf{H},\mathcal{B}'}(\lambda).$$

Claim 3. *For every $s \in \{0, \ldots, q-1\}$ and $i \in [\ell]$, it holds that*

$$\Pr[\mathsf{G}_{1.s.i.B}] = \Pr[\mathsf{G}_{1.s.i+1}].$$

From the above, it holds that

$$
\begin{aligned}
&\left|\Pr[\mathsf{G}_1] - \Pr[\mathsf{G}_2]\right| \\
&= \left|\Pr[\mathsf{G}_{1.0}] - \Pr[\mathsf{G}_{1.q}]\right| \\
&= \left|\sum_{s=0}^{q-1}(\Pr[\mathsf{G}_{1.s}] - \Pr[\mathsf{G}_{1.s+1}])\right| \\
&= \left|\sum_{s=0}^{q-1}(\Pr[\mathsf{G}_{1.s.1}] - \Pr[\mathsf{G}_{1.s.\ell+1}])\right| && (17) \\
&= \left|\sum_{s=0}^{q-1}\sum_{i=1}^{\ell}(\Pr[\mathsf{G}_{1.s.i}] - \Pr[\mathsf{G}_{1.s.i+1}])\right| \\
&= \left|\sum_{s=0}^{q-1}\sum_{i=1}^{\ell}\Big((\Pr[\mathsf{G}_{1.s.i}] - \Pr[\mathsf{G}_{1.s.i.A}]) + (\Pr[\mathsf{G}_{1.s.i.A}] - \Pr[\mathsf{G}_{1.s.i.B}])\right. \\
&\qquad\qquad \left. + (\Pr[\mathsf{G}_{1.s.i.B}] - \Pr[\mathsf{G}_{1.s.i+1}])\Big)\right| \\
&\le \sum_{s=0}^{q-1}\sum_{i=1}^{\ell}\left|\Pr[\mathsf{G}_{1.s.i}] - \Pr[\mathsf{G}_{1.s.i.A}]\right| + \left|\sum_{s=0}^{q-1}\sum_{i=1}^{\ell}(\Pr[\mathsf{G}_{1.s.i.A}] - \Pr[\mathsf{G}_{1.s.i.B}])\right| \\
&\qquad\qquad + \sum_{s=0}^{q-1}\sum_{i=1}^{\ell}\left|\Pr[\mathsf{G}_{1.s.i.B}] - \Pr[\mathsf{G}_{1.s.i+1}]\right| && (18) \\
&\le \frac{q\ell(q\ell-1)}{2\left|\mathcal{R}_{\mathsf{KEM}}\right|} + q\ell\mathsf{Adv}^{\mathsf{ESF}}_{\mathsf{H},\mathcal{B}'}(\lambda), && (19)
\end{aligned}
$$

where Eq. (17) follows from Eqs. (15) and (16), Eq. (18) follows from the triangle inequality, and Eq. (19) follows from Claims 1, 2, and 3. Therefore, if H is entropy-smoothing, then $|\Pr[\mathsf{G}_1] - \Pr[\mathsf{G}_2]|$ is negligible.

To complete the proof, we give the proofs of Claims 1, 2, and 3 in the following.

Proof. (of Claim 1). In this proof, we denote the e-th randomness output from the oracle for the v-th query as $r_{v,e}$ for $v \in \{0, 1, \ldots, q-1\}$ and $e \in [\ell+1]$. Furthermore, we denote the e-th session-key generated in the oracle procedure for the v-th query as $\mathsf{K}_{v,e}$ for $v \in \{0, 1, \ldots, q-1\}$ and $e \in [\ell]$.

For $s \in \{0, \ldots, q-1\}$ and $i \in [\ell]$, Game $1.s.i$ and Game $1.s.i.A$ differ only in the generation of $\mathsf{K}_{q-s,i+1}$. In Game $1.s.i$, $\mathsf{K}_{q-s,i+1}$ is computed as $f(r_{q-s,i})$ using the random function f, whereas in Game $1.s.i.A$, it is sampled uniformly at random from $\{0,1\}^\lambda$. Notably, in both games, $\mathsf{K}_{q-s,e}$ for every $e \in \{j+2, \ldots, \ell+1\}$ is consistently computed as $\mathsf{K}_{q-s,e} = f(r_{q-s,e-1})$.

To prove the lemma, we focus on collisions between $r_{q-s,i}$ and one of the other values that are input to the random function f, namely the values $\{r_{v,e}\}_{v\in[q-s-1],e\in[\ell]}$

and $\{r_{q-s,e}\}_{e\in\{j+1,\ldots,\ell\}}$. (Note that in both Game 1.$s.i$ and Game 1.$s.i.A$, the random function f is never used for the responses to the $(q-s+1)$-th and subsequent queries.) If no such collision occurs, then the distributions of $\mathsf{K}_{q-s,i+1}$ are identical in both Game 1.$s.i$ and Game 1.$s.i.A$, since f is a random function. Hence, all responses returned by the randomness oracle are identically distributed in the two games. However, if a collision occurs, the distributions are no longer identical. To see this, for simplicity, suppose that an adversary always uses the fixed message $00\cdots0$ as its randomness queries. In Game 1.$s.i$, if a collision occurs, namely there exists an index pair $(t,j)\in\{(\upsilon,e)_{\upsilon\in[q-s-1],e\in[\ell]}\cup(q-s,e)_{e\in\{i+1,\ldots,\ell\}}\}$ such that $r_{t,j}=r_{q-s,i}$, then $r_{t,j+1}$ and $r_{q-s,i+1}$ should have the same value. In contrast, in Game 1.$s.i.A$, even if such a collision occurs, they do not necessarily have the same value, since $r_{q-s,i+1}$ and $r_{q-s,i}$ are generated independently in Game 1.$s.i.A$. (More precisely, $r_{q-s,i+1}$ is generated from $\mathsf{K}_{q-s,i+1}$ in both games, but $\mathsf{K}_{q-s,i+1}$ is independent of $r_{q-s,i}$ in Game 1.$s.i.A$.) Consequently, a collision enables $\mathcal{A}$ to distinguish the two games with non-negligible probability by observing the difference in its view.

To formalize this argument, we introduce the collision event. Let $\mathsf{Bad}_{1.s.i}$ be an event in Game 1.$s.i$ that $r_{q-s,i}\in\{r_{\upsilon,e}\}_{\upsilon\in[q-s-1],e\in[\ell]}\cup\{r_{q-s,e}\}_{e\in\{i+1,\ldots,\ell\}}$ occurs during the generation of the response to the $(q-s)$-th query. For Game 1.$s.i.A$, the bad event $\mathsf{Bad}_{1.s.i.A}$ can be defined in the same manner. If $\mathsf{Bad}_{1.s.i}$ does not occur, $\mathsf{K}_{q-s,i}$ can be regarded as a truly random value by the property of a random function. Hence, if the bad events do not occur, $\mathcal{A}$'s view in the two games are identical. Therefore, for every $s\in\{0,\ldots,q-1\}$, and every $i\in[\ell]$, it holds that

$$\Pr[\mathsf{G}_{1.s.i}\wedge\overline{\mathsf{Bad}_{1.s.i}}]=\Pr[\mathsf{G}_{1.s.i.A}\wedge\overline{\mathsf{Bad}_{1.t.j.A}}].$$

Furthermore, $\mathsf{Bad}_{1.s.i}$ and $\mathsf{Bad}_{1.s.i.A}$ can be seen as the same event that a random value $r_{q-s,i}\in\mathcal{R}_{\mathsf{KEM}}$ hits one of the $\ell(q-s-1)+(\ell-i)$ values $\{r_{\upsilon,e}\}_{\upsilon\in[q-s-1],e\in[\ell]}\cup\{r_{q-s,e}\}_{e\in\{i+1,\ldots,\ell\}}$. Therefore, by the union bound, for every $s\in\{0,\ldots,q-1\}$ and $i\in[\ell]$, we have

$$\Pr[\mathsf{Bad}_{1.s.i}]=\Pr[\mathsf{Bad}_{1.s.i.A}]\leq\frac{\ell(q-s)-i}{|\mathcal{R}_{\mathsf{KEM}}|}.$$

Hence, for every $s\in\{0,\ldots,q-1\}$ and $i\in[\ell]$, it holds that

$$\begin{aligned}&\Big|\Pr[\mathsf{G}_{1.s.i}]-\Pr[\mathsf{G}_{1.s.i.A}]\Big|\\&=\Big|\Pr[\mathsf{G}_{1.s.i}\wedge\mathsf{Bad}_{1.s.i}]-\Pr[\mathsf{G}_{1.s.i.A}\wedge\mathsf{Bad}_{1.s.i.A}]\Big|\\&\leq\Pr[\mathsf{Bad}_{1.s.i}]\\&\leq\frac{\ell(q-s)-i}{|\mathcal{R}_{\mathsf{KEM}}|},\end{aligned}$$

which completes the proof. □ (**Claim 1**)

Proof. (of Claim 2). We construct an adversary $\mathcal{B}'$ against the entropy-smoothness of H as follows:

$\mathcal{B}'(k, y)$**:** Firstly, $\mathcal{B}'$ picks $t \in \{0, \ldots, q\}$, and $j \in [\ell]$ uniformly at random. Furthermore, $\mathcal{B}'$ samples $f \in \mathsf{FUNC}_{\mathcal{R}_{\mathsf{KEM}} \to \{0,1\}^\lambda}$ uniformly at random.[10] Then, $\mathcal{B}'$ computes $\mathsf{pp}' \xleftarrow{\$} \mathsf{KSetup}(1^\lambda)$, sets $\mathsf{pp} = (\mathsf{pp}', k)$, and computes $(\mathsf{dpk}, \mathsf{dsk}) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp})$ and $(\mathsf{fpk}_e, \mathsf{fsk}_e) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp})$ for each $e \in [\ell]$. $\mathcal{B}'$ runs $\mathcal{A}(\mathsf{pp}, (\mathsf{fpk}_e)_{e \in [\ell+1]})$. For oracle accesses from $\mathcal{A}$, $\mathcal{B}'$ responds to the first $(q-t-1)$ oracle queries according to Game $1.t.i.B$, and to all oracle queries from the $(q-t+1)$-th onward according to Game $1.s.i.A$. For $\mathcal{A}$'s $(q-t)$-th query $((\mathsf{fm}_e)_{e \in [\ell+1]}, \mathsf{dm})$ to the oracle, $\mathcal{B}'$ executes the following procedures for its responce:

> $\mathcal{B}'$ parses $(\mathsf{ek}, \mathsf{tk}) \leftarrow \mathsf{dpk}$ and $\mathtt{b}_1 \| \mathtt{b}_2 \| \cdots \| \mathtt{b}_\ell \leftarrow \mathsf{dm}$. $\mathcal{B}'$ picks $r_e \in \mathcal{R}_{\mathsf{KEM}}$ uniformly at random for each $e \in [j]$. For generating r_{j+1}, $\mathcal{B}'$ sets $\mathtt{t}_{j+1} \leftarrow y$, and computes $r_{j+1} \leftarrow \mathtt{t}_{j+1} + \mathtt{b}_j \cdot 1_{\mathcal{R}_{\mathsf{KEM}}}$. For generating r_e for each $e \in \{i+2, \ldots, \ell+1\}$, $\mathcal{B}'$ computes $\mathsf{K}_e \leftarrow f(r_{e-1})$, $\mathtt{t}_e \leftarrow \mathsf{H}_k(\mathsf{K}_e)$, and $r_e \leftarrow \mathtt{t}_e + \mathtt{b}_{e-1} \cdot 1_{\mathcal{R}_{\mathsf{KEM}}}$. Finally, $\mathcal{B}'$ returns $(r_e)_{e \in [\ell+1]}$ to $\mathcal{A}$.

Finally, $\mathcal{A}$ returns a bit b' and $\mathcal{B}'$ returns b' as its guess bit.

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

- If y is an output of H, namely y equals to $\mathsf{H}_k(x)$ for randomly chosen x, and it holds that $t = s$ and $j = i$, then $\mathcal{B}'$ simulates Game $1.s.i.A$ perfectly for $\mathcal{A}$, where $y = \mathtt{t}_i$ for the response to the s-th query. Since $\mathcal{B}$ outputs $\mathcal{A}$'s output as it is, we have

$$\Pr_{k \xleftarrow{\$} \mathcal{K}, x \leftarrow \{0,1\}^\lambda} [\mathcal{B}'(k, \mathsf{H}_k(x)) = 1 \mid t = s \wedge j = i] = \Pr[\mathsf{G}_{1.s.i.A}]. \tag{20}$$

- If y is a random value in $\mathcal{R}_{\mathsf{KEM}}$, and it holds that $t = s$ and $j = i$, then $\mathcal{B}'$ simulates Game $1.s.i.B$ perfectly for $\mathcal{A}$. Since $\mathcal{B}$ outputs $\mathcal{A}$'s output as it is, we have

$$\Pr_{k \xleftarrow{\$} \mathcal{K}, y \xleftarrow{\$} \mathcal{R}_{\mathsf{KEM}}} [\mathcal{B}'(k, y) = 1 \mid t = s \wedge j = i] = \Pr[\mathsf{G}_{1.s.i.B}]. \tag{21}$$

[10] Strictly speaking, the PPT algorithm $\mathcal{B}'$ cannot explicitly sample a random function f. Hence, $\mathcal{B}'$ simulates the random function f by lazy sampling.

Hence, we have

$$\begin{aligned}
&\mathsf{Adv}^{\mathtt{ESF}}_{\mathsf{H},\mathcal{B}'}(\lambda)\\
&= \left| \Pr_{k \xleftarrow{\$} \mathcal{K}, x \xleftarrow{\$} \{0,1\}^\lambda} [\mathcal{B}'(k, \mathsf{H}_k(x)) = 1] - \Pr_{k \xleftarrow{\$} \mathcal{K}, y \xleftarrow{\$} \mathcal{R}_{\mathsf{KEM}}} [\mathcal{B}'(k, y) = 1] \right| \\
&= \left| \sum_{t=0}^{q-1} \sum_{j=1}^{\ell} \frac{1}{q\ell} \Big(\Pr_{k \xleftarrow{\$} \mathcal{K}, x \xleftarrow{\$} \{0,1\}^\lambda} [\mathcal{B}'(k, \mathsf{H}_k(x)) = 1 \mid t = s \wedge j = i] \right. \\
&\qquad\qquad \left. - \Pr_{k \xleftarrow{\$} \mathcal{K}, y \xleftarrow{\$} \mathcal{R}_{\mathsf{KEM}}} [\mathcal{B}'(k, y) = 1 \mid t = s \wedge j = i] \Big) \right| \\
&= \frac{1}{q\ell} \left| \sum_{t=0}^{q-1} \sum_{j=1}^{\ell} (\Pr[\mathsf{G}_{1.s.i.A}] - \Pr[\mathsf{G}_{1.s.i.B}]) \right|.
\end{aligned}$$

Therefore, we can conclude

$$\left| \sum_{s=0}^{q-1} \sum_{i=1}^{\ell} (\Pr[\mathsf{G}_{1.s.i.A}] - \Pr[\mathsf{G}_{1.s.i.B}]) \right| = q\ell \mathsf{Adv}^{\mathtt{ESF}}_{\mathsf{H},\mathcal{B}'}(\lambda),$$

which completes the proof. □ (**Claim 2**)

Proof. (of Claim 3). We argue that $\mathcal{A}$'s view in Game $1.s.i.B$ is identical to that in Game $1.s.(i+1)$ for every $s \in \{0, \ldots, q-1\}$ and $i \in [\ell]$. The difference between Game $1.s.i.B$ and Game $1.s.(i+1)$ lies in how r_{i+1} is computed for the $(q-s)$-th query from $\mathcal{A}$. Specifically, in Game $1.s.i.B$, $\mathtt{t}_{i+1} \in \mathcal{R}_{\mathsf{KEM}}$ is sampled uniformly at random, and then r_{i+1} is computed as $r_{i+1} \leftarrow \mathtt{t}_{i+1} + \mathtt{b}_i \cdot 1_{\mathcal{R}_{\mathsf{KEM}}}$, which is also uniformly distributed over $\mathcal{R}_{\mathsf{KEM}}$. On the other hand, in Game $1.s.(i+1)$, r_{i+1} is sampled from $\mathcal{R}_{\mathsf{KEM}}$ directly and uniformly at random. Therefore, the distributions of r_{i+1} in the two games are identical. Hence, $\mathcal{A}$'s views in the two games are identically distributed, and thus, we have

$$\Pr[\mathsf{G}_{1.s.i.B}] = \Pr[\mathsf{G}_{1.s.i+1}],$$

which completes the proof. □ (**Claim 3**)

□ (**Lemma 21**)

□ (**Theorem 13**)

On the other hand, in the setting of S-SAME-R security, an adversary has ek and can use it to test whether the oracle response is random or not since fake randomness of this scheme is generated sequentially using KEMKg with ek. Therefore, the following holds.

Theorem 14. $\mathsf{AME}_{\mathtt{WCHY2}}$ *is not* S-SAME-R *secure. Specifically, there exists a PPT adversary* $\mathcal{A}$ *against the* S-SAME-R *security of* $\mathsf{AME}_{\mathtt{WCHY2}}$ *such that*

$$\mathsf{Adv}^{\mathtt{S\text{-}SAME\text{-}R}}_{\mathsf{AME}_{\mathtt{WCHY2}},\mathcal{A}}(\lambda) = 1 - \frac{1}{|\mathcal{R}_{\mathsf{KEM}}|}.$$

Proof. The description of the adversary $\mathcal{A}$ is as follows:

$\mathcal{A}(\mathsf{pp}, (\mathsf{fpk}_i, \mathsf{fsk}_i)_{i\in[\ell+1]}, \mathsf{dpk})$**:** $\mathcal{A}$ parses $(\mathsf{pp}', k) \leftarrow \mathsf{pp}$ and $(\mathsf{ek}, \mathsf{tk}) \leftarrow \mathsf{dpk}$, samples $\mathsf{fm}_1, \mathsf{fm}_2, \ldots, \mathsf{fm}_{\ell+1}, \mathsf{dm}$ arbitrarily, and sends $((\mathsf{fm}_i)_{i\in[\ell+1]}, \mathsf{dm})$ to the oracle where $\mathsf{dm} = \mathsf{b}_1 \| \cdots \| \mathsf{b}_\ell$ and $\mathsf{b}_i \in \{0,1\}$ for every $i \in [\ell]$. Then, A receives its result $(\mathsf{c}_i, r_i)_{i\in[\ell+1]}$ from the oracle. If $\mathsf{H}_k(\mathsf{KEMKg}(\mathsf{ek}, r_1)) + \mathsf{b}_1 \cdot 1_{\mathcal{R}_{\mathsf{KEM}}} = r_2$, then $\mathcal{A}$ terminates with its output $b' = 0$. Otherwise, $\mathcal{A}$ terminates with its output $b' = 1$.

The above completes the description of $\mathcal{A}$. Let b be the challenge bit. It is not hard to see the following facts:

- If $b = 0$, the probability that $\mathcal{A}$ returns $b' = 0$ is exactly 1 since r_2 is computed by $\mathsf{H}_k(\mathsf{KEMKg}(\mathsf{ek}, r_1)) + \mathsf{b}_1 \cdot 1_{\mathcal{R}_{\mathsf{KEM}}}$ in fRand.
- If $b = 1$, the probability that $\mathcal{A}$ returns $b' = 0$ is $1/|\mathcal{R}_{\mathsf{KEM}}|$ since r_2 is a random value.

Hence, we have

$$\begin{aligned}
&\mathsf{Adv}^{\text{S-SAME-R}}_{\mathsf{AME}_{\text{WCHY2}},\mathcal{A}}(\lambda) \\
&= \Big|\Pr[\mathcal{A}(\mathsf{pp}, (\mathsf{fpk}_i, \mathsf{fsk}_i)_{i\in[\ell+1]}, \mathsf{dpk}) = 0 \mid b = 0] \\
&\qquad\qquad - \Pr[\mathcal{A}(\mathsf{pp}, (\mathsf{fpk}_i, \mathsf{fsk}_i)_{i\in[\ell+1]}, \mathsf{dpk}) = 0 \mid b = 1]\Big| \\
&= 1 - \frac{1}{|\mathcal{R}_{\mathsf{KEM}}|},
\end{aligned}$$

which clearly non-negligible. □ (**Theorem 14**)

Acknowledgement This work was partially supported by JST K Program Grant Number JPMJKP24U3, JST CREST Grant Number JPMJCR22M1, and JSPS KAKENHI Grant Number JP25KJ1176.

References

1. Abdalla, M., Bellare, M., Neven, G.: Robust encryption. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 480–497. Springer, Berlin, Heidelberg (Feb 2010). https://doi.org/10.1007/978-3-642-11799-2_28 57
2. Abdalla, M., Bellare, M., Rogaway, P.: The oracle Diffie-Hellman assumptions and an analysis of DHIES. In: Naccache, D. (ed.) CT-RSA 2001. LNCS, vol. 2020, pp. 143–158. Springer, Berlin, Heidelberg (Apr 2001). https://doi.org/10.1007/3-540-45353-9_12 42
3. Avitabile, G., Botta, V., Giunta, E., Mielniczuk, M., Migliaro, F.: The malice of ELFs: Practical anamorphic-resistant encryption without random oracles. Cryptology ePrint Archive, Report 2025/305 (2025), https://eprint.iacr.org/2025/3058
4. Banerjee, S., Pal, T., Rupp, A., Slamanig, D.: Simple public key anamorphic encryption and signature using multi-message extensions. Cryptology ePrint Archive, Report 2025/370 (2025), https://eprint.iacr.org/2025/370 7, 8

5. Banfi, F., Gegier, K., Hirt, M., Maurer, U., Rito, G.: Anamorphic encryption, revisited. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024, Part II. LNCS, vol. 14652, pp. 3–32. Springer, Cham (May 2024). https://doi.org/10.1007/978-3-031-58723-8_1 7
6. Carnemolla, D., Catalano, D., Giunta, E., Migliaro, F.: Anamorphic resistant encryption: the good, the bad and the ugly. In: Kalai, Y.T., Kamara, S.F. (eds.) CRYPTO 2025, Part III. LNCS, vol. 16002, pp. 472–503. Springer, Cham (Aug 2025). https://doi.org/10.1007/978-3-032-01881-6_15 8
7. Cash, D., Kiltz, E., Shoup, V.: The twin Diffie-Hellman problem and applications. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 127–145. Springer, Berlin, Heidelberg (Apr 2008). https://doi.org/10.1007/978-3-540-78967-3_8 42
8. Catalano, D., Giunta, E., Migliaro, F.: Anamorphic encryption: New constructions and homomorphic realizations. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024, Part II. LNCS, vol. 14652, pp. 33–62. Springer, Cham (May 2024). https://doi.org/10.1007/978-3-031-58723-8_2 7
9. Catalano, D., Giunta, E., Migliaro, F.: Limits of black-box anamorphic encryption. In: Reyzin, L., Stebila, D. (eds.) CRYPTO 2024, Part II. LNCS, vol. 14921, pp. 352–383. Springer, Cham (Aug 2024). https://doi.org/10.1007/978-3-031-68379-4_11 7
10. Catalano, D., Giunta, E., Migliaro, F.: Generic anamorphic encryption, revisited: New limitations and constructions. In: Fehr, S., Fouque, P.A. (eds.) EUROCRYPT 2025, Part II. LNCS, vol. 15602, pp. 275–303. Springer, Cham (May 2025). https://doi.org/10.1007/978-3-031-91124-8_10 7
11. Choi, W., Collins, D., Liu, X., Zikas, V.: A unified treatment of anamorphic encryption. Cryptology ePrint Archive, Report 2025/309 (2025), https://eprint.iacr.org/2025/309 7
12. Chu, H., Do, K., Hanzlik, L., Thyagarajan, S.A.: When threshold meets anamorphic signatures: What is possible and what is not! Cryptology ePrint Archive, Report 2025/1044 (2025), https://eprint.iacr.org/2025/1044 8
13. Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Krawczyk, H. (ed.) CRYPTO'98. LNCS, vol. 1462, pp. 13–25. Springer, Berlin, Heidelberg (Aug 1998). https://doi.org/10.1007/BFb0055717 42
14. Deo, A., Libert, B.: Anamorphic signatures with dictator and recipient unforgeability for long messages. In: ASIACRYPT 2025, Part VI. pp. 370–401. LNCS, Springer, Singapore (Dec 2025). https://doi.org/10.1007/978-981-95-5119-4_12 8
15. Deo, A., Libert, B.: Fully asymmetric anamorphic homomorphic encryption from LWE. Cryptology ePrint Archive, Report 2025/328 (2025), https://eprint.iacr.org/2025/328 7
16. Dodis, Y., Goldin, E.: Anamorphic-resistant encryption; or why the encryption debate is still alive. In: Kalai, Y.T., Kamara, S.F. (eds.) CRYPTO 2025, Part III. LNCS, vol. 16002, pp. 440–471. Springer, Cham (Aug 2025). https://doi.org/10.1007/978-3-032-01881-6_14 7
17. GANGULY, S., Chaudhury, S.S.: Computational quantum anamorphic encryption and anamorphic secret sharing. Cryptology ePrint Archive, Report 2025/399 (2025), https://eprint.iacr.org/2025/399 7
18. Hanaoka, G., Kurosawa, K.: Efficient chosen ciphertext secure public key encryption under the computational Diffie-Hellman assumption. In: Pieprzyk, J. (ed.)

ASIACRYPT 2008. LNCS, vol. 5350, pp. 308–325. Springer, Berlin, Heidelberg (Dec 2008). https://doi.org/10.1007/978-3-540-89255-7_19 42

19. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM Journal on Computing **28**(4), 1364–1396 (1999) 30, 71
20. Hofheinz, D., Kiltz, E.: Secure hybrid encryption from weakened key encapsulation. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 553–571. Springer, Berlin, Heidelberg (Aug 2007). https://doi.org/10.1007/978-3-540-74143-5_31 42
21. Jaeger, J., Stracovsky, R.: Dictators? Friends? Forgers. - breaking and fixing unforgeability definitions for anamorphic signature schemes. In: Chung, K.M., Sasaki, Y. (eds.) ASIACRYPT 2024, Part II. LNCS, vol. 15485, pp. 105–137. Springer, Singapore (Dec 2024). https://doi.org/10.1007/978-981-96-0888-1_4 8
22. Kiltz, E.: Chosen-ciphertext secure key-encapsulation based on gap hashed Diffie-Hellman. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 282–297. Springer, Berlin, Heidelberg (Apr 2007). https://doi.org/10.1007/978-3-540-71677-8_19 42
23. Kurosawa, K., Desmedt, Y.: A new paradigm of hybrid encryption scheme. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 426–442. Springer, Berlin, Heidelberg (Aug 2004). https://doi.org/10.1007/978-3-540-28628-8_26 42
24. Kutylowski, M., Persiano, G., Phan, D.H., Yung, M., Zawada, M.: Anamorphic signatures: Secrecy from a dictator who only permits authentication! In: Handschuh, H., Lysyanskaya, A. (eds.) CRYPTO 2023, Part II. LNCS, vol. 14082, pp. 759–790. Springer, Cham (Aug 2023). https://doi.org/10.1007/978-3-031-38545-2_25 8
25. Kutylowski, M., Persiano, G., Phan, D.H., Yung, M., Zawada, M.: The self-anti-censorship nature of encryption: On the prevalence of anamorphic cryptography. PoPETs **2023**(4), 170–183 (Oct 2023). https://doi.org/10.56553/popets-2023-0104 7
26. Persiano, G., Phan, D.H., Yung, M.: Anamorphic encryption: Private communication against a dictator. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022, Part II. LNCS, vol. 13276, pp. 34–63. Springer, Cham (May / Jun 2022). https://doi.org/10.1007/978-3-031-07085-3_2 3, 4, 5, 6, 7, 13, 14, 16, 17
27. Persiano, G., Phan, D.H., Yung, M.: Public-key anamorphism in (CCA-secure) public-key encryption and beyond. In: Reyzin, L., Stebila, D. (eds.) CRYPTO 2024, Part II. LNCS, vol. 14921, pp. 422–455. Springer, Cham (Aug 2024). https://doi.org/10.1007/978-3-031-68379-4_13 7
28. Wang, W., Han, S., Liu, S.: Anamorphic authenticated key exchange: Double key distribution under surveillance. In: Chung, K.M., Sasaki, Y. (eds.) ASIACRYPT 2024, Part V. LNCS, vol. 15488, pp. 168–200. Springer, Singapore (Dec 2024). https://doi.org/10.1007/978-981-96-0935-2_6 7
29. Wang, Y., Chen, R., Huang, X., Yung, M.: Sender-anamorphic encryption reformulated: Achieving robust and generic constructions. In: Guo, J., Steinfeld, R. (eds.) ASIACRYPT 2023, Part VI. LNCS, vol. 14443, pp. 135–167. Springer, Singapore (Dec 2023). https://doi.org/10.1007/978-981-99-8736-8_5 3, 4, 5, 6, 13, 14, 18, 19, 20, 39, 40, 56, 57, 58

A Proofs of Lemmas in Section 2

A.1 Proof of Lemma 1

Let $\mathcal{D}_0 := \{h = 0;\ \text{do } \{h \leftarrow h+1;\ x_h \xleftarrow{\$} X\} \text{ until } \mathsf{P}(x_h) = 1 : x_h\}$, and $\mathcal{D}_1 := \{x \xleftarrow{\$} \mathsf{P}^{-1}(1) : x\}$. Here, we show that $\mathcal{D}_0 \equiv \mathcal{D}_1$. Since P^{-1} partitions X, we will analyze $\Pr[\mathcal{D}_0 = x']$ and $\Pr[\mathcal{D}_1 = x']$ for any $x' \in \mathsf{P}^{-1}(0)$ and $x' \in \mathsf{P}^{-1}(1)$ respectively.

For any $x' \in \mathsf{P}^{-1}(0)$, it is easy to see that $\Pr[\mathcal{D}_0 = x'] = \Pr[\mathcal{D}_1 = x'] = 0$ by the definitions of $\mathcal{D}_0$ and $\mathcal{D}_1$.

For any $x' \in \mathsf{P}^{-1}(1)$, $\Pr[\mathcal{D}_1 = x'] = 1/|\mathsf{P}^{-1}(1)|$ since $\mathcal{D}_1$ is the uniform distribution over $\mathsf{P}^{-1}(1)$, and

$$
\begin{aligned}
&\Pr[\mathcal{D}_0 = x'] \\
&= \Pr[\mathsf{P}(x_1) = 1 \wedge x_1 = x'] + \Pr[\mathsf{P}(x_1) = 0 \wedge \mathsf{P}(x_2) = 1 \wedge x_2 = x'] \\
&\quad + \Pr[\mathsf{P}(x_1) = 0 \wedge \mathsf{P}(x_2) = 0 \wedge \mathsf{P}(x_3) = 1 \wedge x_3 = x'] + \cdots && (22)\\
&= \sum_{h=1}^{\infty} \Pr\Big[\bigwedge_{u=1}^{h-1} \big(\mathsf{P}(x_u) = 0\big) \wedge \mathsf{P}(x_h) = 1 \wedge x_h = x'\Big] && (23)\\
&= \sum_{h=1}^{\infty} \Pr\Big[\bigwedge_{u=1}^{h-1} \big(\mathsf{P}(x_u) = 0\big) \wedge x_h = x'\Big] && (24)\\
&= \sum_{h=1}^{\infty} \Big(1 - \frac{|\mathsf{P}^{-1}(1)|}{|X|}\Big)^{h-1} \cdot \frac{1}{|X|} && (25)\\
&= \frac{1}{|X|} \cdot \frac{1}{1 - (1 - \frac{|\mathsf{P}^{-1}(1)|}{|X|})} && (26)\\
&= \frac{1}{|\mathsf{P}^{-1}(1)|},
\end{aligned}
$$

where $x_1, x_2, \ldots$ are sampled uniformly at random from X in Eqs. (22), (23), and (24); Eq. (22) holds due to the definition of $\mathcal{D}_0$; Eq. (24) holds since $\mathsf{P}(x') = 1$ is implied by $x_i = x' \in \mathsf{P}^{-1}(1)$; Eq. (25) holds since $x_1, x_2, \ldots$ are chosen independently; and Eq. (26) holds since it is an infinite geometric series whose first term is $1/|X|$ and whose common ratio is $(1 - |\mathsf{P}^{-1}(1)|/|X|)$. Therefore, we can see that, for any $x' \in \mathsf{P}^{-1}(1)$, $\Pr[\mathcal{D}_0 = x'] = 1/|\mathsf{P}^{-1}(1)| = \Pr[\mathcal{D}_0 = x']$.

From the above, for any $x' \in X$, we have $\Pr[\mathcal{D}_0 = x'] = \Pr[\mathcal{D}_1 = x']$. Namely, it holds that $\mathcal{D}_0 \equiv \mathcal{D}_1$. □ (**Lemma 1**)

A.2 Proof of Lemma 2

Let $\mathcal{D}_0 := \{x \xleftarrow{\$} X; y \leftarrow f(x) : (x, y)\}$, and $\mathcal{D}_1 := \{x \xleftarrow{\$} X; y \leftarrow f(x); \widehat{x} \xleftarrow{\$} f^{-1}(y) : (\widehat{x}, y)\}$. It is sufficient to show $\mathcal{D}_0 \equiv \mathcal{D}_1$, namely for any $x' \in X, y' \in Y$, it holds that $\Pr[\mathcal{D}_0 = (x', y')] = \Pr[\mathcal{D}_1 = (x', y')]$.

For any $(x', y') \in X \times Y$ such that $y' \neq f(x')$, we have

$$\Pr[\mathcal{D}_0 = (x', y')] = \Pr[\mathcal{D}_1 = (x', y')] = 0.$$

For any $(x', y') \in X \times Y$ such that $y' = f(x')$, the following holds:

$$\begin{aligned}\Pr[\mathcal{D}_0 = (x', y')] &= \Pr[(x, y) \xleftarrow{\$} \mathcal{D}_0 : x = x' \wedge f(x) = f(x')] \\ &= \Pr[(x, y) \xleftarrow{\$} \mathcal{D}_0 : x = x'] \qquad (27) \\ &= \frac{1}{|X|},\end{aligned}$$

where Eq. (27) holds since $x = x'$ implies $f(x) = f(x')$. Furthermore, the following also holds:

$$\begin{aligned}&\Pr[\mathcal{D}_1 = (x', y')] \\ &= \Pr[(\widehat{x}, y) \xleftarrow{\$} \mathcal{D}_1 : \widehat{x} = x' \wedge f(x) = f(x')] \qquad (28) \\ &= \Pr[(\widehat{x}, y) \xleftarrow{\$} \mathcal{D}_1 : f(x) = f(x')] \cdot \Pr[(\widehat{x}, y) \xleftarrow{\$} \mathcal{D}_1 : \widehat{x} = x' \mid f(x) = f(x')] \\ &= \Pr[(\widehat{x}, y) \xleftarrow{\$} \mathcal{D}_1 : x \in f^{-1}(f(x'))] \cdot \frac{1}{|f^{-1}(f(x'))|} \qquad (29) \\ &= \frac{|f^{-1}(f(x'))|}{|X|} \cdot \frac{1}{|f^{-1}(f(x'))|} \\ &= \frac{1}{|X|},\end{aligned}$$

Eq. (28) holds due to the definition of $\mathcal{D}_1$, and Eq. (29) holds since $\widehat{x}$ has been chosen uniformly at random from $f^{-1}(f(x))$ conditioned on $f(x) = f(x')$. Therefore, we can see that, for any $(x', y') \in X \times Y$ such that $y' = f(x')$,

$$\Pr[\mathcal{D}_0 = (x', y')] = \frac{1}{|X|} = \Pr[\mathcal{D}_1 = (x', y')].$$

From the above, for any $(x', y') \in X \times Y$, we have

$$\Pr[\mathcal{D}_0 = (x', y')] = \Pr[\mathcal{D}_1 = (x', y')].$$

Namely, it holds that $\mathcal{D}_0 \equiv \mathcal{D}_1$. □ (**Lemma 2**)

B Proof of Theorem 2 in Section 3

The implication that SAME-R (resp. S-SAME-R security) implies SAME-R′ (resp. S-SAME-R′) security is trivial since an adversary considered in the former notion receives more information than one in the latter notion.

We now show that if an ℓ-SAME scheme AME is SAME-R′ secure, then AME is also SAME-R secure. Let $\mathcal{A}$ be an SAME-R adversary. Using $\mathcal{A}$ as a building block, we construct a reduction algorithm $\mathcal{B}$ as follows:

$\mathcal{B}(\mathsf{pp}, (\mathsf{fpk}_i)_{i\in[\ell]})$**:** $\mathcal{B}$ runs $\mathcal{A}(\mathsf{pp}, (\mathsf{fpk}_i)_{i\in[\ell]})$. When $\mathcal{A}$ queries $((\mathsf{fm}_i)_{i\in[\ell]}, \mathsf{dm})$ to the encryption oracle, $\mathcal{B}$ queries $((\mathsf{fm}_i)_{i\in[\ell]}, \mathsf{dm})$ to the randomness oracle, receives the response $(r_i)_{i\in[\ell]}$ from the randomness oracle, computes $\mathsf{c}_i \leftarrow \mathsf{Enc}(\mathsf{fpk}_i, \mathsf{fm}_i; r_i)$ for every $i \in [\ell]$, and sends $((\mathsf{c}_i)_{i\in[\ell]}, (r_i)_{i\in[\ell]})$ to $\mathcal{A}$ as the response to the encryption query. Finally, $\mathcal{B}$ returns $\mathcal{A}$'s output.

$\mathcal{B}$ computes $(\mathsf{c}_i)_{i\in[\ell]}$ in exactly the same way as how they are computed in the encryption oracle of the SAME-R security experiment. Therefore, $\mathcal{B}$ simulates the SAME-R security experiment perfectly for $\mathcal{A}$. Furthermore, $\mathcal{B}$ uses the output of $\mathcal{A}$ as is. Thus, it holds that

$$\mathsf{Adv}^{\texttt{SAME-R}}_{\mathsf{AME},\mathcal{A}}(\lambda) = \mathsf{Adv}^{\texttt{SAME-R}'}_{\mathsf{AME},\mathcal{B}}(\lambda) = \mathsf{negl}(\lambda).$$

Therefore, we can conclude that AME is SAME-R secure if and only if AME is SAME-R′ secure. The implication that S-SAME-R′ security implies S-SAME-R security is established analogously. □ (**Theorem 2**)

C Robustness

In this section, we review the definition of robustness and discuss the robustness of the (ℓ-)SAME schemes in this paper.

Robustness is an important property for SAME, and it was introduced by Wang et al. [29]. Roughly speaking, robustness captures two properties: (1) when the anamorphic decryption algorithm takes a normal (non-anamorphic) ciphertext as input, it outputs the special symbol $\perp$ indicating a failure of decryption, and (2) when the anamorphic decryption algorithm takes a secret key and the anamorphic ciphertext to which the secret key does not correspond, it outputs $\perp$.

Formally, the definition of robustness is as follows.

Definition 14 (Robustness of SAME). *We say that an ℓ-SAME scheme* AME *is robust if for any PPT adversaries $\mathcal{A}_0$ and $\mathcal{A}_1$, the following probabilities are negligible:*

$$\Pr\left[\begin{array}{ll} \mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda); & \\ \forall i \in [\ell],\ (\mathsf{fpk}_i, \mathsf{fsk}_i) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp}); & \\ (\mathsf{dpk}, \mathsf{dsk}) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp}); & : \mathsf{dDec}(\mathsf{dsk}, (\mathsf{c}_i)_{i\in[\ell]}) \neq \perp \\ \mathsf{FM} \xleftarrow{\$} \mathcal{A}_0(\mathsf{pp}, \mathsf{FPK}); & \\ (r_i)_{i\in[\ell]} \xleftarrow{\$} \mathcal{R}^\ell; & \\ \forall i \in [\ell], \mathsf{c}_i \leftarrow \mathsf{Enc}(\mathsf{fpk}_i, \mathsf{fm}_i; r_i) & \end{array}\right],$$

and

$$\Pr\left[\begin{array}{l}\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda);\\ \forall i \in [\ell],\ (\mathsf{fpk}_i, \mathsf{fsk}_i) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp});\\ (\mathsf{dpk}, \mathsf{dsk}) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp});\\ (\mathsf{dpk}', \mathsf{dsk}') \xleftarrow{\$} \mathsf{KG}(\mathsf{pp});\\ (\mathsf{FM}, \mathsf{dm}) \xleftarrow{\$} \mathcal{A}_1(\mathsf{pp}, \mathsf{FPK}, \mathsf{dpk});\\ (r_i)_{i\in[\ell]} \xleftarrow{\$} \mathsf{fRand}(\mathsf{pp}, \mathsf{FPK}, \mathsf{FM}, \mathsf{dpk}, \mathsf{dm});\\ \forall i \in [\ell], \mathsf{c}_i \leftarrow \mathsf{Enc}(\mathsf{fpk}_i, \mathsf{fm}_i; r_i)\end{array} : \mathsf{dDec}(\mathsf{dsk}', (\mathsf{c}_i)_{i\in[\ell]}) \neq \bot\right],$$

where FPK *denotes* $(\mathsf{fpk}_i)_{i\in[\ell]}$*, and* FM *denotes* $(\mathsf{fm}_i)_{i\in[\ell]}$*.*

C.1 Robustness of Each Scheme

Here, we discuss whether each SAME scheme satisfies robustness.

Persiano et al.'s SAME Scheme The SAME scheme $\mathsf{AME}_{\mathtt{PPY}}$ proposed by Persiano et al. does not inherently guarantee robustness. This stems from the fact that its anamorphic decryption algorithm is identical to that of the underlying PKE scheme. If the PKE decryption algorithm is not designed to output $\bot$, the anamorphic decryption algorithm inherits this property. For instance, an instantiation using ElGamal encryption, which satisfies the three security conditions, never outputs $\bot$ during decryption. Consequently, $\mathsf{AME}_{\mathtt{PPY}}$ does not guarantee robustness.

Wang et al.'s First SAME Scheme Wang et al. showed that their first scheme $\mathsf{AME}_{\mathtt{WCHY1}}$ satisfies robustness assuming that the underlying PKE scheme satisfies pseudorandomness and robustness, and H is entropy-smoothing [29, Theorem 4.1]. Here, we recall that the formal definition of robustness of PKE is following.[11]

Definition 15 (Robustness of PKE [1]). *We say that a PKE scheme* PKE *is robust if for any PPT adversary* $\mathcal{A}$*, the following probability is negligible:*

$$\Pr\left[\begin{array}{l}\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda);\\ (\mathsf{pk}_0, \mathsf{sk}_0) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp});\\ (\mathsf{pk}_1, \mathsf{sk}_1) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp});\\ \mathsf{m} \xleftarrow{\$} \mathcal{A}(\mathsf{pp}, \mathsf{pk}_0, \mathsf{pk}_1);\\ \mathsf{c} \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}_0, \mathsf{m});\\ \mathsf{m}^* \leftarrow \mathsf{Dec}(\mathsf{sk}_1, \mathsf{c})\end{array} : \mathsf{m}^* \neq \bot\right].$$

[11] This definition is simplified from the original definition. Since these two definitions are equivalent up to polynomial security loss, we use the simplified one.

Modified Wang et al.'s First SAME Scheme To show the robustness of the modified Wang et al.'s first scheme $\mathsf{AME}_{\texttt{M-WCHY1}}$ presented in Section 5.2, we first observe that its basic structure is identical to that of $\mathsf{AME}_{\texttt{WCHY1}}$. Therefore, it suffices to show that each building block used in $\mathsf{AME}_{\texttt{M-WCHY1}}$ satisfies the corresponding properties required in the robustness proof of $\mathsf{AME}_{\texttt{WCHY1}}$. More specifically, it is sufficient to show that the hybrid encryption scheme composed of a KEM scheme and a DEM scheme satisfies pseudorandomness and robustness as a PKE scheme, and H is entropy-smoothing.

First, pseudorandomness of the hybrid encryption scheme follows directly from the pseudorandomness of the underlying KEM and DEM schemes.

Next, in $\mathsf{AME}_{\texttt{M-WCHY1}}$, H is assumed to be a PRF and is not necessarily an entropy-smoothing function. Nevertheless, by Lemma 12, a PRF provides an analogous property sufficient for the robustness proof for $\mathsf{AME}_{\texttt{M-WCHY1}}$. Therefore, the game transition corresponding to the step in the proof for $\mathsf{AME}_{\texttt{WCHY1}}$ can be carried out similarly.

It remains to show robustness of the hybrid encryption scheme. This property holds under either of the following conditions: (1) the KEM scheme is robust; or (2) the KEM scheme is IND-CPA secure (implied by pseudorandomness) and the DEM scheme is robust.[12] For condition (1), if the KEM scheme is robust, decapsulating a session-key under a different secret key results in $\perp$ with overwhelming probability, in which case the decryption of the hybrid encryption scheme also fails. Therefore, both of the probabilities in the definition of the robustness for SAME are negligible. For condition (2), by the robustness of the DEM scheme, the adversary can succeed (with non-negligible probability) only if the session-key generated from pk_0 coincides with the session-key obtained by decapsulating the KEM ciphertext using sk_1. Such a collision occurs only with negligible probability due to the IND-CPA security of the KEM scheme.

As a conclusion, if we assume that the KEM scheme or the DEM scheme is robust in addition to the assumptions for the SAME-R security proof, we can show that $\mathsf{AME}_{\texttt{M-WCHY1}}$ is robust.

Wang et al.'s Second SAME Scheme Wang et al. showed that their second scheme, denoted by $\mathsf{AME}_{\texttt{WCHY2}}$, is robust [29, Theorem 5.1]. To show this result, they assume that the underlying KEM scheme satisfies key-pseudorandomness, universal decryptability, and homomorphic property[13], and H is entropy-smoothing.

D Proofs of Theorem 6 and Lemmas in Section 5

D.1 Proof of Theorem 6

Let $(\mathsf{fm}_i)_{i\in[\ell]} \in (\{0,1\}^n)^\ell$ be ℓ forced messages, and let $\mathsf{dm} \in \{0,1\}^n$ be a duplicate message. Let $\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$, let $(\mathsf{fpk}_i, \mathsf{fsk}_i)_{i\in[\ell]}$ and $(\mathsf{dpk}, \mathsf{dsk})$ be

[12] Robustness of KEM and DEM can be defined similarly to that of PKE, and thus we omit their formal definitions.

[13] For the formal definitions for these notions, see [29].

generated by $\mathsf{KG}(\mathsf{pp})$, let $(r_i)_{i\in[\ell]} \xleftarrow{\$} \mathsf{fRand}(\mathsf{pp}, (\mathsf{fpk}_i)_{i\in[\ell]}, (\mathsf{fm}_i)_{i\in[\ell]}, \mathsf{dpk}, \mathsf{dm})$, and let $\mathsf{c}_i \leftarrow \mathsf{Enc}(\mathsf{fpk}_i, \mathsf{fm}_i; r_i)$ for every $i \in [\ell]$. To show the correctness of $\mathsf{AME}_{\mathtt{WCHY1}}$, we need to show that $\Pr[\mathsf{dDec}(\mathsf{dsk}, (\mathsf{c}_i)_{i\in[\ell]}) \neq \mathsf{dm}]$ is negligible, where the probability takes the generation of the above values into account.

Let Bad be the event that, in the process of running $\mathsf{fRand}(\mathsf{pp}, (\mathsf{fpk}_i)_{i\in[\ell]}, (\mathsf{fm}_i)_{i\in[\ell]}, \mathsf{dpk}, \mathsf{dm})$, the "do-until" loop for some $i \in [\ell]$ terminates because the counter $\mathtt{ctr}$ reaches λ before the condition $\mathsf{b}^* = \mathsf{b}_i$ is satisfied. Formally, let Bad be the event that the condition "$\mathsf{b}^* \neq \mathsf{b}_i \wedge \mathtt{ctr} \geq \lambda$" holds in some i-th iteration of "for" loop for some $i \in [\ell]$. Note that if Bad does not occur, $\mathsf{c}' = \mathsf{b}_1||\mathsf{b}_2||\cdots||\mathsf{b}_\ell$ is an encryption of dm under dpk where $\mathsf{b}_i = \mathsf{H}_k(\mathsf{Enc}'(\mathsf{fpk}_i, \mathsf{fm}_i; r_i))$ for every $i \in [\ell]$. Hence, it holds that $\Pr[\mathsf{dDec}(\mathsf{dsk}, (\mathsf{c}_i)_{i\in[\ell]}) \neq \mathsf{dm} \wedge \overline{\mathsf{Bad}}] = 0$ by the correctness of PKE, since $\mathsf{dDec}(\mathsf{dsk}, (\mathsf{c}_i)_{i\in[\ell]}) = \mathsf{Dec}'(\mathsf{dsk}, \mathsf{c}') = \mathsf{dm}$. Therefore, we can see that $\Pr[\mathsf{dDec}(\mathsf{dsk}, (\mathsf{c}_i)_{i\in[\ell]}) \neq \mathsf{dm}] \leq \Pr[\mathsf{Bad}]$ holds.

Hence, it remains to show that $\Pr[\mathsf{Bad}] \leq \mathsf{negl}(\lambda)$. We show this by relying on Lemma 3. More specifically, we show that there exists a PPT adversary $\mathcal{A}$ that satisfies $\Pr[\mathsf{Bad}] \leq \ell\mathsf{Adv}^{\mathtt{Fail}}_{\mathsf{PKE},\mathsf{H},\mathcal{A}}(\lambda)$, where $\mathsf{Adv}^{\mathtt{Fail}}_{\mathsf{PKE},\mathsf{H},\mathcal{A}}(\lambda)$ is the advantage of an adversary $\mathcal{A}$ in the experiment $\mathsf{Expt}^{\mathtt{Fail}}_{\mathsf{PKE},\mathsf{H},\mathcal{A}}(\lambda)$ defined in Lemma 3.

For $i \in [\ell]$, let Bad_i be the event that the condition "$\mathsf{b}^* \neq \mathsf{b}_i \wedge \mathtt{ctr} \geq \lambda$" holds in the i-th iteration of "for" loop. By the union bound, it holds that $\Pr[\mathsf{Bad}] \leq \sum_{i\in[\ell]} \Pr[\mathsf{Bad}_i]$.

Now, consider an adversary $\mathcal{A}$ in Lemma 3, which, on input $(k, \mathsf{pp}, \mathsf{pk})$, picks $j \in [\ell]$ uniformly at random, and returns fm_j. For such $\mathcal{A}$, $\mathsf{Expt}^{\mathtt{Fail}}_{\mathsf{PKE},\mathsf{H},\mathcal{A}}(\lambda) = 1$ holds when $\mathsf{H}_k(\mathsf{Enc}(\mathsf{fpk}_j, \mathsf{fm}_j; r_h)) \neq \mathsf{H}_k(\mathsf{Enc}(\mathsf{fpk}_j, \mathsf{fm}_j; r_{\lambda+1}))$ for every $h \in [\lambda]$, where $r_1, \ldots, r_{\lambda+1} \xleftarrow{\$} \mathcal{R}^{\lambda+1}$. Since this is precisely the same condition that causes Bad_j, we have $\mathsf{Adv}^{\mathtt{Fail}}_{\mathsf{PKE},\mathsf{H},\mathcal{A}}(\lambda) = (1/\ell)\sum_{i\in[\ell]} \Pr[\mathsf{Bad}_i]$. Hence, we have

$$\Pr[\mathsf{Bad}] \leq \ell\mathsf{Adv}^{\mathtt{Fail}}_{\mathsf{PKE},\mathsf{H},\mathcal{A}}(\lambda),$$

which is negligible due to Lemma 3, the pseudorandomness of PKE, and the entropy-smoothiness of H.

From the above,

$$\begin{aligned}
&\Pr[\mathsf{dDec}(\mathsf{dsk}, (\mathsf{c}_i)_{i\in[\ell]}) \neq \mathsf{dm}] \\
&= \Pr[\mathsf{dDec}(\mathsf{dsk}, (\mathsf{c}_i)_{i\in[\ell]}) \neq \mathsf{dm} \wedge \mathsf{Bad}] \\
&\qquad + \Pr[\mathsf{dDec}(\mathsf{dsk}, (\mathsf{c}_i)_{i\in[\ell]}) \neq \mathsf{dm} \wedge \overline{\mathsf{Bad}}] \\
&\leq \ell\mathsf{Adv}^{\mathtt{Fail}}_{\mathsf{PKE},\mathsf{H},\mathcal{A}}(\lambda) \\
&\leq \mathsf{negl}(\lambda).
\end{aligned}$$

□ (**Theorem 6**)

D.2 Proof of Lemma 3

We will show that for any PPT adversary $\mathcal{A}$, there exist PPT adversaries $\mathcal{B}$ and $\mathcal{B}'$ such that

$$\Pr[\mathsf{Expt}^{\mathsf{Fail}}_{\mathsf{PKE},\mathsf{H},\mathcal{A}}(\lambda) = 1] \leq \lambda\big(\mathsf{Adv}^{\mathtt{PR}}_{\mathsf{PKE},\mathcal{B}}(\lambda) + \mathsf{Adv}^{\mathtt{ESF}}_{\mathsf{H},\mathcal{B}'}(\lambda)\big) + 2^{-\lambda}.$$

We define the following sequence of two games for $\mathcal{A}$. Let G_N be the event that $\mathcal{A}$ returns 1 in Game $N \in \{0,1\}$.

Game 0: This is exactly $\mathsf{Expt}^{\mathsf{Fail}}_{\mathsf{PKE},\mathsf{H},\mathcal{A}}(\lambda)$. Hence,

$$\Pr[\mathsf{Expt}^{\mathsf{Fail}}_{\mathsf{PKE},\mathsf{H},\mathcal{A}}(\lambda) = 1] = \Pr[\mathsf{G}_0].$$

Game 1: This game is the same as Game 0 except that $\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_\lambda$ are chosen uniformly at random from $\{0,1\}^\ell$.

Firstly, we will show that there exists a PPT adversary $\mathcal{B}$ that satisfies $|\Pr[\mathsf{G}_0] - \Pr[\mathsf{G}_1]| = \lambda \mathsf{Adv}^{\mathsf{PR}}_{\mathsf{PKE},\mathcal{B}}(\lambda)$. Using $\mathcal{A}$ as a building block, we construct a reduction algorithm $\mathcal{B} = (\mathcal{B}_0, \mathcal{B}_1)$ attacking the pseudorandomness of PKE as follows:

$\mathcal{B}_0(\mathsf{pp}, \mathsf{pk})$**:** $\mathcal{B}_0$ picks $k \in \mathcal{K}$ uniformly at random, runs $\mathsf{m} \xleftarrow{\$} \mathcal{A}(k, \mathsf{pp}, \mathsf{pk})$, sets $\mathrm{st}_{\mathcal{B}}$ as all the values known to $\mathcal{B}_0$, and terminates with output $(\mathsf{m}, \mathrm{st}_{\mathcal{B}})$.

$\mathcal{B}_1(\mathsf{c}, \mathrm{st}_{\mathcal{B}})$**:** $\mathcal{B}_1$ picks $u \in [\lambda]$ uniformly at random, and computes $\mathsf{c}_{\lambda+1} \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$ and $\mathsf{b}_{\lambda+1} \leftarrow \mathsf{H}_k(\mathsf{c}_{\lambda+1})$. Then, $\mathcal{B}_1$ generates c_h for each $h \in [\lambda]$ as follows:

- If $h \in [u-1]$, then $\mathcal{B}_1$ picks $\mathsf{c}_h \in \{0,1\}^\ell$ uniformly at random, and computes $\mathsf{b}_h \leftarrow \mathsf{H}_k(\mathsf{c}_h)$.
- If $h = u$, then $\mathcal{B}_1$ sets $\mathsf{c}_u \leftarrow \mathsf{c}$ and $\mathsf{b}_u \leftarrow \mathsf{H}_k(\mathsf{c}_u)$.
- If $h \in \{u+1, \ldots, \lambda\}$, then $\mathcal{B}_1$ computes $\mathsf{c}_h \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$ and $\mathsf{b}_h \leftarrow \mathsf{H}_k(\mathsf{c}_h)$.

Finally, if $\mathsf{b}_h \neq \mathsf{b}_{\lambda+1}$ holds for every $h \in [\lambda]$, then $\mathcal{B}_1$ returns $\beta' = 1$. Otherwise, $\mathcal{B}_1$ returns $\beta' = 0$.

The above completes the description of $\mathcal{B}$. Let β be $\mathcal{B}$'s challenge bit. It is not hard to see the following facts.

- If $\beta = 0$ and $u = 1$, then $\mathcal{B}$ simulates Game 0 perfectly for $\mathcal{A}$. Therefore, the probability that $\mathcal{B}$ returns the guess bit $\beta' = 1$ is equivalent to $\Pr[\mathsf{G}_0]$, namely,
$$\Pr[\beta' = 1 \mid \beta = 0 \wedge u = 1] = \Pr[\mathsf{G}_0]. \tag{30}$$
- If $\beta = 1$ and $u = \lambda$, then $\mathcal{B}$ simulates Game 1 perfectly for $\mathcal{A}$. Therefore, the probability that $\mathcal{B}$ returns the guess bit $\beta' = 1$ is equivalent to $\Pr[\mathsf{G}_1]$, namely,
$$\Pr[\beta' = 1 \mid \beta = 1 \wedge u = \lambda] = \Pr[\mathsf{G}_1]. \tag{31}$$
- For any $v \in [\lambda - 1]$, the games simulated by $\mathcal{B}$ in each of the following situations are identical:
  - $\mathcal{B}$'s challenge bit β is 1 and $u = v$; and
  - $\mathcal{B}$'s challenge bit β is 0 and $u = v + 1$.

  Hence, the probabilities in these situations are identical, namely for any $v \in [\lambda - 1]$,
$$\Pr[\beta' = 1 \mid \beta = 1 \wedge u = v] = \Pr[\beta' = 1 \mid \beta = 0 \wedge u = v + 1]. \tag{32}$$

Hence, we can compute $\mathcal{B}$'s advantage against the pseudorandomness of PKE as follows:

$$\begin{aligned}
\mathsf{Adv}^{\mathtt{PR}}_{\mathsf{PKE},\mathcal{B}}(\lambda) &= \big|\Pr[\beta' = 1 \mid \beta = 0] - \Pr[\beta' = 1 \mid \beta = 1]\big| \\
&= \Big|\sum_{v\in[\lambda]} \Pr[u = v]\big(\Pr[\beta' = 1 \mid \beta = 0 \wedge u = v] \\
&\qquad\qquad - \Pr[\beta' = 1 \mid \beta = 1 \wedge u = v]\big)\Big| \\
&= \frac{1}{\lambda}\Big|\Pr[\beta' = 1 \mid \beta = 0 \wedge u = 1] - \Pr[\beta' = 1 \mid \beta = 1 \wedge u = \lambda]\Big| \qquad (33) \\
&= \frac{1}{\lambda}\Big|\Pr[\mathsf{G}_0] - \Pr[\mathsf{G}_1]\Big|, \qquad (34)
\end{aligned}$$

where Eq. (33) follows from Eq. (32), and Eq. (34) follows from Eqs. (30) and (31). Therefore, the following holds:

$$\Big|\Pr[\mathsf{G}_0] - \Pr[\mathsf{G}_1]\Big| = \lambda\mathsf{Adv}^{\mathtt{PR}}_{\mathsf{PKE},\mathcal{B}}(\lambda).$$

Next, we show that there exists a PPT adversary $\mathcal{B}'$ such that

$$\Big|\Pr[\mathsf{G}_1] - 2^{-\lambda}\Big| = \lambda\mathsf{Adv}^{\mathtt{ESF}}_{\mathsf{H},\mathcal{B}'}(\lambda).$$

Using $\mathcal{A}$ as a building block, we construct a reduction algorithm $\mathcal{B}'$ attacking the entropy-smoothness of H as follows:

$\mathcal{B}'(k, y)$**:** $\mathcal{B}'$ computes $\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$ and $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp})$, runs $\mathsf{m} \xleftarrow{\$} \mathcal{A}(k, \mathsf{pp}, \mathsf{pk})$, and computes $\mathsf{c}_{\lambda+1} \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$ and $\mathtt{b}_{\lambda+1} \leftarrow \mathsf{H}_k(\mathsf{c}_{\lambda+1})$. Then, $\mathcal{B}'$ picks $u \in [\lambda]$ uniformly at random, and generates $\mathtt{b}_h$ for each $h \in [\lambda]$ as follows:

- If $h \in [u-1]$, then $\mathcal{B}'$ picks $\mathtt{b}_h \in \{0,1\}$ uniformly at random.
- If $h = u$, then $\mathcal{B}'$ sets $\mathtt{b}_u \leftarrow y$.
- If $h \in \{u+1, \ldots, \lambda\}$, then $\mathcal{B}'$ picks $\mathsf{c}_h \in \{0,1\}^\ell$ uniformly at random, and computes $\mathtt{b}_h \leftarrow \mathsf{H}_k(\mathsf{c}_h)$.

Finally, if $\mathtt{b}_h \neq \mathtt{b}_{\lambda+1}$ holds for every $h \in [\lambda]$, then $\mathcal{B}'$ returns 1. Otherwise, $\mathcal{B}'$ returns 0.

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

- If y is an output of $\mathsf{H}_k(x)$ where $x \xleftarrow{\$} \{0,1\}^\ell$, and $u = 1$, then $\mathcal{B}'$ simulates Game 1 perfectly for $\mathcal{A}$ so that $\mathsf{c}_u = x$ is chosen uniformly at random from $\{0,1\}^\ell$. Therefore, the probability that $\mathcal{B}'$ returns 1 is equivalent to $\Pr[\mathsf{G}_1]$, namely,
$$\Pr_{k,x}[\mathcal{B}'(k, \mathsf{H}_k(x)) = 1 \mid u = 1] = \Pr[\mathsf{G}_1], \qquad (35)$$
where $k \in \mathcal{K}$ and $x \in \{0,1\}^\ell$ are chosen uniformly at random.

– If y is a random bit and $u = \lambda$, then the probability that $\mathcal{B}'$ returns 1 is equal to $2^{-\lambda}$ since it is the probability that $\mathtt{b}_h \neq \mathtt{b}_{\lambda+1}$ holds for every $h \in [\lambda]$ where $\mathtt{b}_h \xleftarrow{\$} \{0,1\}$. Hence,

$$\Pr_{k,y}[\mathcal{B}'(k,y) = 1 \mid u = \lambda] = 2^{-\lambda}, \tag{36}$$

where $k \in \mathcal{K}$ and $y \in \{0,1\}$ are chosen uniformly at random.
– For any $v \in [\lambda - 1]$, the experiments simulated by $\mathcal{B}'$ in each of the following situations are identical:
  • y is an outputs of $\mathsf{H}_k(x)$ where $x \xleftarrow{\$} \{0,1\}^\ell$, and $u = v + 1$; and
  • y is a random bit and $u = v$.

  Hence, the probabilities in these situations are identical, namely for any $v \in [\lambda - 1]$,

$$\Pr_{k,x}[\mathcal{B}'(k,\mathsf{H}_k(x)) = 1 \mid u = v+1] = \Pr_{k,y}[\mathcal{B}'(k,y) = 1 \mid u = v], \tag{37}$$

where $k \in \mathcal{K}, x \in \{0,1\}^\ell$ and $y \in \{0,1\}$ are chosen uniformly at random.

Hence, we can compute $\mathcal{B}'$'s advantage against the entropy-smoothness of H as follows:

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{ESF}}_{\mathsf{H},\mathcal{B}'}(\lambda) &= \left|\Pr_{k,x}[\mathcal{B}'(k,\mathsf{H}_k(x)) = 1] - \Pr_{k,y}[\mathcal{B}'(k,y) = 1]\right| \\
&= \Bigg|\sum_{v\in[\lambda]} \Pr[u = v]\Big(\Pr_{k,x}[\mathcal{B}'(k,\mathsf{H}_k(x)) = 1 \mid u = v] \\
&\qquad\qquad - \Pr_{k,y}[\mathcal{B}'(k,y) = 1 \mid u = v]\Big)\Bigg| \\
&= \frac{1}{\lambda}\left|\Pr_{k,x}[\mathcal{B}'(k,\mathsf{H}_k(x)) = 1 \mid u = 1] - \Pr_{k,y}[\mathcal{B}'(k,y) = 1 \mid u = \lambda]\right| \qquad (38) \\
&= \frac{1}{\lambda}\left|\Pr[\mathsf{G}_1] - 2^{-\lambda}\right|, \qquad (39)
\end{aligned}$$

where $k \in \mathcal{K}, x \in \{0,1\}^\ell$ and $y \in \{0,1\}$ are chosen uniformly at random, Eq. (38) follows from Eq. (37), and Eq. (39) follows from Eqs. (35) and (36). Therefore, the following holds:

$$\left|\Pr[\mathsf{G}_1] - 2^{-\lambda}\right| = \lambda \mathsf{Adv}^{\mathsf{ESF}}_{\mathsf{H},\mathcal{B}'}(\lambda).$$

From the above and the triangle inequality, the following holds:

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{Fail}}_{\mathsf{PKE},\mathsf{H},\mathcal{A}}(\lambda) &= \Pr[\mathsf{G}_0] \\
&\leq \left|\Pr[\mathsf{G}_0] - \Pr[\mathsf{G}_1]\right| + \left|\Pr[\mathsf{G}_1] - 2^{-\lambda}\right| + 2^{-\lambda} \\
&= \lambda\Big(\mathsf{Adv}^{\mathsf{PR}}_{\mathsf{PKE},\mathcal{B}}(\lambda) + \mathsf{Adv}^{\mathsf{ESF}}_{\mathsf{H},\mathcal{B}'}(\lambda)\Big) + 2^{-\lambda}.
\end{aligned}$$

Therefore, if PKE satisfies pseudorandomness and H is entropy-smoothing, then $\mathsf{Adv}^{\mathsf{Fail}}_{\mathsf{PKE},\mathsf{H},\mathcal{A}}(\lambda)$ is negligible. □ (**Lemma 3**)

D.3 Proof of Lemma 4

We will show that there exists a PPT adversary $\mathcal{B}$ such that $|\Pr[\mathsf{G}_0] - \Pr[\mathsf{G}_1]| = q\mathsf{Adv}^{\mathsf{PR}}_{\mathsf{PKE},\mathcal{B}}(\lambda)$. Using $\mathcal{A}$ as a building block, we construct a reduction algorithm $\mathcal{B} = (\mathcal{B}_0, \mathcal{B}_1)$ attacking the pseudorandomness of PKE as follows:

$\mathcal{B}_0(\mathsf{pp}', \mathsf{pk})$**:** $\mathcal{B}_0$ picks $k \in \mathcal{K}$ and $t \in [q]$ uniformly at random, sets $\mathsf{dpk} \leftarrow \mathsf{pk}$ and $\mathsf{pp} \leftarrow (\mathsf{pp}', k)$, and computes $(\mathsf{fpk}_i, \mathsf{fsk}_i) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp})$ for each $i \in [\ell]$. $\mathcal{B}_0$ runs $\mathcal{A}(\mathsf{pp}, (\mathsf{fpk}_i)_{i\in[\ell]})$ and executes the following procedure for answering $\mathcal{A}$'s s-th query $((\mathsf{fm}_i)_{i\in[\ell]}, \mathsf{dm})$ to the oracle where $s \in [t]$:

- If $s < t$, then $\mathcal{B}_0$ computes $(r_i)_{i\in[\ell]}$ in the same way as $\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game1}}$, and returns $(r_i)_{i\in[\ell]}$ to $\mathcal{A}$.
- If $s = t$, then $\mathcal{B}_0$ sets st as all the values known to $\mathcal{B}_0$, and terminates with output $(\mathsf{dm}, \mathrm{st})$.

$\mathcal{B}_1(\mathsf{c}', \mathrm{st})$**:** $\mathcal{B}_1$ executes the following procedure and returns its result $(r_i)_{i\in[\ell]}$ to $\mathcal{A}$ as its answer to $\mathcal{A}$'s t-th query.

$\mathtt{b}_1 \| \mathtt{b}_2 \| \cdots \| \mathtt{b}_\ell \leftarrow \mathsf{c}'$
for $i \in [\ell]$
  $\mathtt{ctr} \leftarrow 0$
  do
    $r^* \xleftarrow{\$} \mathcal{R}$
    $\mathsf{c}^* \leftarrow \mathsf{Enc}'(\mathsf{fpk}_i, \mathsf{fm}_i; r^*)$
    $\mathsf{b}^* \leftarrow \mathsf{H}_k(\mathsf{c}^*)$
    $\mathtt{ctr} \leftarrow \mathtt{ctr} + 1$
  until $\mathsf{b}^* = \mathtt{b}_i \vee \mathtt{ctr} \geq \lambda$
  $r_i \leftarrow r^*$

For the s-th query to the oracle from $\mathcal{A}$ where $s > t$, $\mathcal{B}_1$ executes the procedure of $\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game0}}$ and returns its result $(r_i)_{i\in[\ell]}$ to $\mathcal{A}$.
Finally, $\mathcal{A}$ returns a bit b' and $\mathcal{B}_1$ returns $\beta' = b'$.

The above completes the description of $\mathcal{B}$. Let β be $\mathcal{B}$'s challenge bit. It is not hard to see the following facts.

- If $\beta = 0$ and $t = 1$, then $\mathcal{B}$ simulates Game 0 perfectly for $\mathcal{A}$. Therefore, the probability that $\mathcal{B}$ returns $\beta' = 1$ is identical to the probability that $\mathcal{A}$ returns 1 in Game 0, namely,
$$\Pr[\beta' = 1 \mid \beta = 0 \wedge t = 1] = \Pr[\mathsf{G}_0]. \tag{40}$$
- If $\beta = 1$ and $t = q$, then $\mathcal{B}$ simulates Game 1 perfectly for $\mathcal{A}$. Therefore, the probability that $\mathcal{B}$ returns $\beta' = 1$ is identical to the probability that $\mathcal{A}$ returns 1 in Game 1, namely,
$$\Pr[\beta' = 1 \mid \beta = 1 \wedge t = q] = \Pr[\mathsf{G}_1]. \tag{41}$$
- From $\mathcal{A}$'s viewpoint, for any $s \in [q-1]$, the experiments simulated by $\mathcal{B}$ in each of the following situations are identical:

- $\beta = 1$ and $t = s$; and
- $\beta = 0$ and $t = s + 1$.

Since $\mathcal{B}$ outputs $\mathcal{A}$'s output as it is, for every $s \in [q-1]$, we have

$$\Pr[\beta' = 1 \mid \beta = 1 \wedge t = s] = \Pr[\beta' = 1 \mid \beta = 0 \wedge t = s+1]. \tag{42}$$

Hence, we can compute $\mathcal{B}$'s advantage against the pseudorandomness of PKE as follows:

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{PR}}_{\mathsf{PKE},\mathcal{B}}(\lambda) &= \Big|\Pr[\beta' = 1 \mid \beta = 0] - \Pr[\beta' = 1 \mid \beta = 1]\Big| \\
&= \Big|\sum_{s\in[q]} \Pr[t = s]\Big(\Pr[\beta' = 1 \mid \beta = 0 \wedge t = s] \\
&\qquad\qquad - \Pr[\beta' = 1 \mid \beta = 1 \wedge t = s]\Big)\Big| \\
&= \frac{1}{q}\Big|\Pr[\beta' = 1 \mid \beta = 0 \wedge t = 1] - \Pr[\beta' = 1 \mid \beta = 1 \wedge t = q]\Big| \qquad (43) \\
&= \frac{1}{q}\Big|\Pr[\mathsf{G}_0] - \Pr[\mathsf{G}_1]\Big|. \qquad (44)
\end{aligned}$$

Eq. (43) follows from Eq. (42), and Eq. (44) follows from Eqs. (40) and (41). This completes the proof. □ (**Lemma 4**)

D.4 Proof of Lemma 6

We will show that there exists a PPT adversary $\mathcal{B}'$ such that $|\Pr[\mathsf{G}_2] - \Pr[\mathsf{G}_3]| = q\ell\mathsf{Adv}^{\mathsf{ESF}}_{\mathsf{H},\mathcal{B}'}(\lambda)$. Using $\mathcal{A}$ as a building block, we construct a reduction algorithm $\mathcal{B}'$ attacking the entropy-smoothness of H as follows:

$\mathcal{B}'(k, y)$**:** $\mathcal{B}'$ picks $t \in [q]$ and $j \in [\ell]$ uniformly at random, generates a public parameter $\mathsf{pp}' \xleftarrow{\$} \mathsf{Setup}'(1^\lambda)$, and sets $\mathsf{pp} \leftarrow (\mathsf{pp}', k)$. Furthermore, $\mathcal{B}'$ generates forced key pairs $(\mathsf{fpk}_i, \mathsf{fsk}_i) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp})$ for each $i \in [\ell]$. $\mathcal{B}'$ runs $\mathcal{A}(\mathsf{pp}, (\mathsf{fpk}_i)_{i\in[\ell]})$ and executes the following procedure for $\mathcal{A}$'s s-th query $((\mathsf{fm}_i)_{i\in[\ell]}, \mathsf{dm})$ to the oracle where $s \in [q]$:

- If $s < t$, then $\mathcal{B}'$ computes $(r_i)_{i\in[\ell]}$ in the same way as $\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game3}}$, and returns $(r_i)_{i\in[\ell]}$ to $\mathcal{A}$.
- If $s = t$, then $\mathcal{B}'$ executes the following procedure and returns its result $(r_i)_{i\in[\ell]}$ to $\mathcal{A}$:

for $i \in [\ell]$
  if $i \leq j-1$ then
    $\widehat{\mathsf{c}}_i \xleftarrow{\$} \{0,1\}^\ell$
    $\mathtt{b}_i \leftarrow \mathsf{H}_k(\widehat{\mathsf{c}}_i)$
  else if $i = j$ then
    $\mathtt{b}_j \leftarrow y$
  else if $i \geq j+1$ then
    $\mathtt{b}_i \xleftarrow{\$} \{0,1\}$
  $\mathtt{ctr} \leftarrow 0$
  do
    $r^* \xleftarrow{\$} \mathcal{R}$
    $\mathsf{c}^* \leftarrow \mathsf{Enc}'(\mathsf{fpk}_i, \mathsf{fm}_i; r^*)$
    $\mathtt{b}^* \leftarrow \mathsf{H}_k(\mathsf{c}^*)$
    $\mathtt{ctr} \leftarrow \mathtt{ctr} + 1$
  until $\mathtt{b}^* = \mathtt{b}_i \vee \mathtt{ctr} \geq \lambda$
  $r_i \leftarrow r^*$

  - If $s > t$, then $\mathcal{B}'$ computes $(r_i)_{i\in[\ell]}$ in the same way as $\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game2}}$, and returns its output $(r_i)_{i\in[\ell]}$ to $\mathcal{A}$.

Finally, $\mathcal{A}$ returns a bit b' and $\mathcal{B}'$ returns b' as its own guess bit.

The above completes the description of $\mathcal{B}'$. It is not hard to see the following facts where $k \in \mathcal{K}, \mathsf{c} \in \{0,1\}^\ell, \mathtt{b} \in \{0,1\}$ are chosen uniformly at random.

- If $\mathcal{B}'$'s input y is a random value $\mathtt{b}$, $t = 1$, and $j = 1$, then $\mathcal{B}'$ simulates Game 2 perfectly for $\mathcal{A}$. Therefore, the probability that $\mathcal{B}'$ returns 1 is identical to the probability that $\mathcal{A}$ returns 1 in Game 2, namely,

$$\Pr_{k,\mathtt{b}}[\mathcal{B}'(k,\mathtt{b}) = 1 \mid t = 1 \wedge j = 1] = \Pr[\mathsf{G}_2]. \tag{45}$$

- If $\mathcal{B}'$'s input y is an output of H, $t = q$, and $j = \ell$, then $\mathcal{B}'$ simulates Game 3 perfectly for $\mathcal{A}$. Therefore, the probability that $\mathcal{B}'$ returns 1 is identical to the probability that $\mathcal{A}$ returns 1 in Game 3, namely,

$$\Pr_{k,\mathsf{c}}[\mathcal{B}'(k,\mathsf{H}_k(\mathsf{c})) = 1 \mid t = q \wedge j = \ell] = \Pr[\mathsf{G}_3]. \tag{46}$$

- From $\mathcal{A}$'s viewpoint, for every $s \in [q]$ and $i \in [\ell-1]$, the experiments simulated by $\mathcal{B}'$ in each of the following situations are identical:
  - $\mathcal{B}'$'s input y is an output of H, $t = s$, and $j = i$; and
  - $\mathcal{B}'$'s input y is a random value $\mathtt{b}$, $t = s$, and $j = i+1$.

  Since $\mathcal{B}'$ outputs $\mathcal{A}$'s output as it is, for every $s \in [q]$ and $i \in [\ell-1]$, we have

$$\Pr_{k,\mathsf{c}}[\mathcal{B}'(k,\mathsf{H}_k(\mathsf{c})) = 1 \mid t = s \wedge j = i] = \Pr_{k,\mathtt{b}}[\mathcal{B}'(k,\mathtt{b}) = 1 \mid t = s \wedge j = i+1]. \tag{47}$$

- From $\mathcal{A}$'s viewpoint, for every $s \in [q-1]$, the experiments simulated by $\mathcal{B}'$ in each of the following situations are identical:

- $\mathcal{B}'$'s input is an output of H, $t = s$, and $j = \ell$; and
- $\mathcal{B}'$'s input is a random value b, $t = s+1$, and $j = 1$.

Since $\mathcal{B}'$ outputs $\mathcal{A}$'s output as it is, for every $s \in [q-1]$, we have

$$\Pr_{k,\mathsf{c}}[\mathcal{B}'(k, \mathsf{H}_k(\mathsf{c})) = 1 \mid t = s \wedge j = \ell] = \Pr_{k,\mathsf{b}}[\mathcal{B}'(k, \mathsf{b}) = 1 \mid t = s+1 \wedge j = 1]. \quad (48)$$

Hence, we can compute $\mathcal{B}'$'s advantage against the entropy-smoothness of H as follows:

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{ESF}}_{\mathsf{H},\mathcal{B}'}(\lambda) &= \left|\Pr_{k,\mathsf{c}}[\mathcal{B}'(k, \mathsf{H}_k(\mathsf{c})) = 1] - \Pr_{k,\mathsf{b}}[\mathcal{B}'(k, \mathsf{b}) = 1]\right| \\
&= \Bigg|\sum_{s\in[q]}\sum_{i\in[\ell]} \Pr_{t,j}[t = s \wedge j = i]\Big(\Pr_{k,\mathsf{c}}[\mathcal{B}'(k, \mathsf{H}_k(\mathsf{c}')) = 1 \mid t = s \wedge j = i] \\
&\qquad\qquad - \Pr_{k,\mathsf{b}}[\mathcal{B}'(k, \mathsf{b}) = 1 \mid t = s \wedge j = i]\Big)\Bigg| \\
&= \Bigg|\sum_{s\in[q]} \Pr_{t}[t = s] \sum_{i\in[\ell]} \Pr_{j}[j = i]\Big(\Pr_{k,\mathsf{c}}[\mathcal{B}'(k, \mathsf{H}_k(\mathsf{c}')) = 1 \mid t = s \wedge j = i] \\
&\qquad\qquad - \Pr_{k,\mathsf{b}}[\mathcal{B}'(k, \mathsf{b}) = 1 \mid t = s \wedge j = i]\Big)\Bigg| \\
&= \frac{1}{q\ell}\Big|\Pr_{k,\mathsf{c}}[\mathcal{B}'(k, \mathsf{H}_k(\mathsf{c})) = 1 \mid t = q \wedge j = \ell] \\
&\qquad\qquad - \Pr_{k,\mathsf{b}}[\mathcal{B}'(k, \mathsf{b}) = 1 \mid t = 1 \wedge j = 1]\Big| \qquad (49) \\
&= \frac{1}{q\ell}\Big|\Pr[\mathsf{G}_3] - \Pr[\mathsf{G}_2]\Big|, \qquad (50)
\end{aligned}$$

where $k \in \mathcal{K}, \mathsf{c} \in \{0,1\}^\ell, y \in \{0,1\}, t \in [q]$, and $j \in [\ell]$ are chosen uniformly at random. Eq. (49) follows from Eqs. (47) and (48), and Eq. (50) follows from Eqs. (45) and (46). This completes the proof. □ (**Lemma 6**)

D.5 Proof of Lemma 7

We will show that there exists a PPT adversary $\mathcal{B}''$ such that $|\Pr[\mathsf{G}_3] - \Pr[\mathsf{G}_4]| = q\ell\mathsf{Adv}^{\mathsf{PR}}_{\mathsf{PKE},\mathcal{B}''}(\lambda)$. Using $\mathcal{A}$ as a building block, we construct a reduction algorithm $\mathcal{B}'' = (\mathcal{B}''_0, \mathcal{B}''_1)$ attacking the pseudorandomness of PKE as follows:

$\mathcal{B}''_0(\mathsf{pp}', \mathsf{pk})$**:** $\mathcal{B}''_0$ picks $k \in \mathcal{K}$, $t \in [q]$, and $j \in [\ell]$ uniformly at random, sets $\mathsf{fpk}_j \leftarrow \mathsf{pk}$ and $\mathsf{pp} = (\mathsf{pp}', k)$, and computes $(\mathsf{fpk}_i, \mathsf{fsk}_i) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp})$ for each $i \in [\ell]\backslash\{j\}$. $\mathcal{B}''$ runs $\mathcal{A}(\mathsf{pp}, (\mathsf{fpk}_i)_{i\in[\ell]})$ and executes the following procedure for $\mathcal{A}$'s s-th query $((\mathsf{fm}_i)_{i\in[\ell]}, \mathsf{dm})$ to the oracle where $s \in [q]$:

- If $s < t$, then $\mathcal{B}''_0$ computes $(r_i)_{i\in[\ell]}$ in the same way as $\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game4}}$, and returns $(r_i)_{i\in[\ell]}$ to $\mathcal{A}$.
- If $s = t$, then $\mathcal{B}''_0$ sets st as all the values known to $\mathcal{B}''_0$, and terminates its execution with its output $(\mathsf{fm}_j, \mathrm{st})$.

$\mathcal{B}''_1(\mathsf{c}, \mathrm{st})$**:** $\mathcal{B}''_1$ executes the following procedure and returns its result $(r_i)_{i\in[\ell]}$ to $\mathcal{A}$ as its answer to $\mathcal{A}$'s t-th query.

for $i \in [\ell]$
  if $i \le j-1$ then
    $\widehat{r}_i \xleftarrow{\$} \mathcal{R}$
    $\widehat{\mathsf{c}}_i \leftarrow \mathsf{Enc}'(\mathsf{fpk}_i, \mathsf{fm}_i; \widehat{r}_i)$
  else if $i = j$ then
    $\widehat{\mathsf{c}}_j \leftarrow \mathsf{c}$
  else if $i \ge j+1$ then
    $\widehat{\mathsf{c}}_i \xleftarrow{\$} \{0,1\}^\ell$
  $\mathsf{b}_i \leftarrow \mathsf{H}_k(\widehat{\mathsf{c}}_i)$
  $\mathtt{ctr} \leftarrow 0$
  do
    $r^* \xleftarrow{\$} \mathcal{R}$
    $\mathsf{c}^* \leftarrow \mathsf{Enc}(\mathsf{fpk}_i, \mathsf{fm}_i; r^*)$
    $\mathsf{b}^* \leftarrow \mathsf{H}_k(\mathsf{c}^*)$
    $\mathtt{ctr} \leftarrow \mathtt{ctr} + 1$
  until $\mathsf{b}^* = \mathsf{b}_i \vee \mathtt{ctr} \ge \lambda$
  $r_i \leftarrow r^*$

For the s-th query to the oracle from $\mathcal{A}$ where $s > t$, $\mathcal{B}''_1$ executes the procedure of $\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game3}}$ and returns its result $(r_i)_{i\in[\ell]}$ to $\mathcal{A}$.
Finally, $\mathcal{A}$ returns a bit b' and $\mathcal{B}''_1$ returns $\beta' = b'$.

The above completes the description of $\mathcal{B}''$. Let β be $\mathcal{B}''$'s challenge bit. It is not hard to see the following facts.

- If $\beta = 1$, (namely, $\mathcal{B}''$'s input is a random value,) and $t = 1$ and $j = 1$, then $\mathcal{B}''$ simulates Game 3 perfectly for $\mathcal{A}$. Therefore, the probability that $\mathcal{B}''$ returns $\beta' = 1$ is identical to the probability that $\mathcal{A}$ returns $b' = 1$ in Game 3, namely,
$$\Pr[\beta' = 1 \mid \beta = 1 \wedge t = 1 \wedge j = 1] = \Pr[\mathsf{G}_3]. \tag{51}$$
- If $\beta = 0$, (namely, $\mathcal{B}''$'s input is an encryption of fm_j,) and $t = q$ and $j = \ell$, then $\mathcal{B}''$ simulates Game 4 perfectly for $\mathcal{A}$. Therefore, the probability that $\mathcal{B}''$ returns $\beta = 1$ is identical to the probability that $\mathcal{A}$ returns $b' = 1$ in Game 4, namely,
$$\Pr[\beta' = 1 \mid \beta = 0 \wedge t = q \wedge j = \ell] = \Pr[\mathsf{G}_4]. \tag{52}$$
- From $\mathcal{A}$'s viewpoint, for any $s \in [q]$ and $i \in [\ell-1]$, the experiments simulated by $\mathcal{B}''$ in each of the following situations are identical:
  - $\beta = 0$, $t = s$, and $j = i$; and
  - $\beta = 1$, $t = s$, and $j = i+1$.

  Since $\mathcal{B}''$ outputs $\mathcal{A}$'s output as it is, for every $s \in [q]$ and $i \in [\ell-1]$, we have
$$\Pr[\beta' = 1 \mid \beta = 0 \wedge t = s \wedge j = i] = \Pr[\beta' = 1 \mid \beta = 1 \wedge t = s \wedge j = i+1]. \tag{53}$$

– From $\mathcal{A}$'s viewpoint, for any $s \in [q-1]$, the experiments simulated by $\mathcal{B}''$ in each of the following situations are identical:
  - $b = 0$, $t = s$, and $j = \ell$; and
  - $b = 1$, $t = s+1$, and $j = 1$.

  Since $\mathcal{B}''$ outputs $\mathcal{A}$'s output as it is, for every $s \in [q-1]$, we have

$$\Pr[\beta' = 1 \mid \beta = 0 \wedge t = s \wedge j = \ell] = \Pr[\beta' = 1 \mid \beta = 1 \wedge t = s+1 \wedge j = 1]. \quad (54)$$

Hence, we can compute $\mathcal{B}''$'s advantage against pseudorandomness of PKE as follows:

$$\begin{aligned}
\mathsf{Adv}^{\mathtt{PR}}_{\mathsf{PKE},\mathcal{B}''}(\lambda) &= \Big|\Pr[\beta' = 1 \mid \beta = 0] - \Pr[\beta' = 1 \mid \beta = 1]\Big| \\
&= \Big|\sum_{s\in[q]}\sum_{i\in[\ell]} \Pr_{t,j}[j = s \wedge j = i]\Big(\Pr[\beta' = 1 \mid \beta = 0 \wedge t = s \wedge j = i] \\
&\qquad - \Pr[\beta' = 1 \mid \beta = 1 \wedge t = s \wedge j = i]\Big)\Big| \\
&= \Big|\sum_{s\in[q]} \Pr_{t}[t = s] \sum_{i\in[\ell]} \Pr_{j}[j = i]\Big(\Pr[\beta' = 1 \mid \beta = 0 \wedge t = s \wedge j = i] \\
&\qquad - \Pr[\beta' = 1 \mid \beta = 1 \wedge t = s \wedge j = i]\Big)\Big| \\
&= \frac{1}{q\ell}\Big|\Pr[\beta' = 1 \mid \beta = 0 \wedge t = q \wedge j = \ell] \\
&\qquad - \Pr[\beta' = 1 \mid \beta = 1 \wedge t = 1 \wedge j = 1]\Big| \qquad (55) \\
&= \frac{1}{q\ell}\Big|\Pr[\mathsf{G}_4] - \Pr[\mathsf{G}_3]\Big|, \qquad (56)
\end{aligned}$$

where $t \in [q], j \in [\ell]$ are chosen uniformly at random, Eq. (55) follows from Eqs. (53) and (54), and Eq. (56) follows from Eqs. (51) and (52). This completes the proof. □ (**Lemma 7**)

D.6 Proof of Lemma 11

We will show that there exist PPT adversaries $\mathcal{B}$ and $\mathcal{B}'$ such that

$$\begin{aligned}
\epsilon &:= \Pr\left[\begin{array}{ll} k \xleftarrow{\$} \mathcal{K}; & \\ \mathsf{pp}' \xleftarrow{\$} \mathsf{KSetup}(1^\lambda); & \\ (\mathsf{pk},\mathsf{sk}) \xleftarrow{\$} \mathsf{KKG}(\mathsf{pp}'); & : \forall h \in [\lambda], \mathsf{b}_h \neq \mathsf{b}_{\lambda+1} \\ \forall h \in [\lambda+1], (\mathsf{c}_h, \mathsf{K}_h) \xleftarrow{\$} \mathsf{Encap}(\mathsf{pk}); & \\ \forall h \in [\lambda+1], \mathsf{b}_h \leftarrow \mathsf{H}_k(\mathsf{c}_h) & \end{array}\right] \\
&\leq (\lambda+1)\mathsf{Adv}^{\mathtt{CTX\text{-}PR}}_{\mathsf{KEM},\mathcal{B}}(\lambda) + \mathsf{Adv}^{\mathtt{PRF}}_{\mathsf{H},\mathcal{B}'}(\lambda) + \lambda(\lambda+1)\cdot 2^{-(\ell_{\mathsf{KEM}}+1)} + 2^{-\lambda}.
\end{aligned}$$

To show the lemma, we consider the following probability ϵ':

$$\epsilon' := \Pr\left[\begin{array}{ll} k \xleftarrow{\$} \mathcal{K}; & \\ \forall h \in [\lambda+1], \mathsf{c}_h \xleftarrow{\$} \{0,1\}^{\ell_{\mathsf{KEM}}}; & : \forall h \in [\lambda], \mathsf{b}_h \neq \mathsf{b}_{\lambda+1} \\ \forall h \in [\lambda+1], \mathsf{b}_h \leftarrow \mathsf{H}_k(\mathsf{c}_h) & \end{array}\right].$$

By the triangle inequality, $\epsilon \leq |\epsilon - \epsilon'| + \epsilon'$ holds.

First, we will show that there exists a PPT adversary $\mathcal{B}$ such that $|\epsilon - \epsilon'| = (\lambda+1)\mathsf{Adv}^{\mathtt{CTX\text{-}PR}}_{\mathsf{KEM},\mathcal{B}}(\lambda)$. To show this, we consider the following probability for each $u \in \{0,\dots,\lambda+1\}$:

$$\epsilon_u := \Pr\left[\begin{array}{ll} k \xleftarrow{\$} \mathcal{K}; & \\ \mathsf{pp}' \xleftarrow{\$} \mathsf{KSetup}(1^\lambda); & \\ (\mathsf{pk},\mathsf{sk}) \xleftarrow{\$} \mathsf{KKG}(\mathsf{pp}'); & \\ \forall h \in \{0,\dots,u\}, \mathsf{c}_h \xleftarrow{\$} \{0,1\}^{\ell_{\mathsf{KEM}}}; & : \forall h \in [\lambda], \mathsf{b}_h \neq \mathsf{b}_{\lambda+1} \\ \forall h \in \{u+1,\dots,\lambda+1\}, \mathsf{c}_h \xleftarrow{\$} \mathsf{Encap}(\mathsf{pk}); & \\ \forall h \in [\lambda+1], \mathsf{b}_h \leftarrow \mathsf{H}_k(\mathsf{c}_h) & \end{array}\right].$$

By the definition of ϵ_u, $\epsilon_0 = \epsilon$ and $\epsilon_{\lambda+1} = \epsilon'$. Hence, we have

$$|\epsilon - \epsilon'| = |\epsilon_0 - \epsilon_{\lambda+1}| = \Big|\sum_{u\in[\lambda+1]}(\epsilon_{u-1} - \epsilon_u)\Big|$$

by the triangle inequality.

Here, to show the existence of $\mathcal{B}$ such that $|\sum_{u\in[\lambda+1]}(\epsilon_{u-1} - \epsilon_u)| = (\lambda+1)\mathsf{Adv}^{\mathtt{CTX\text{-}PR}}_{\mathsf{KEM},\mathcal{B}}(\lambda)$, we construct a reduction algorithm $\mathcal{B}$ against the ciphertext pseudorandomness of KEM as follows:

$\mathcal{B}(\mathsf{pp},\mathsf{pk},\mathsf{c})$**:** $\mathcal{B}$ picks $k \in \mathcal{K}$ and $u \in [\lambda+1]$ uniformly at random, sets $\mathsf{c}_u \leftarrow \mathsf{c}$, samples $\mathsf{c}_h \xleftarrow{\$} \{0,1\}^{\ell_{\mathsf{KEM}}}$ for every $h \in \{0,\dots,u-1\}$, computes $(\mathsf{c}_h, \mathsf{K}_h) \xleftarrow{\$} \mathsf{Encap}(\mathsf{pk})$ for every $h \in \{u+1,\dots,\lambda+1\}$, and computes $\mathsf{b}_h \leftarrow \mathsf{H}_k(\mathsf{c}_h)$ for every $h \in [\lambda+1]$. If $\mathsf{b}_h \neq \mathsf{b}_{\lambda+1}$ for every $h \in [\lambda]$, then $\mathcal{B}$ returns 1. Otherwise, $\mathcal{B}$ returns 0.

The above completes the description of $\mathcal{B}$. If c is a ciphertext of KEM, it is easy to see that $\mathcal{B}$ perfectly simulates the operations for defining ϵ_{u-1}. If c is a random value, it is easy to see that $\mathcal{B}$ perfectly simulates the operations for defining ϵ_u. Therefore, it holds that

$$\mathsf{Adv}^{\mathtt{CTX\text{-}PR}}_{\mathsf{KEM},\mathcal{B}}(\lambda) = \frac{1}{\lambda+1}\Big|\sum_{u\in[\lambda+1]}(\epsilon_{u-1} - \epsilon_u)\Big|,$$

namely, we have

$$\Big|\sum_{u\in[\lambda+1]}(\epsilon_{u-1} - \epsilon_u)\Big| = (\lambda+1)\mathsf{Adv}^{\mathtt{CTX\text{-}PR}}_{\mathsf{KEM},\mathcal{B}}(\lambda).$$

Hence, we have

$$\big|\epsilon - \epsilon'\big| = (\lambda+1)\mathsf{Adv}^{\mathtt{CTX\text{-}PR}}_{\mathsf{KEM},\mathcal{B}}(\lambda). \tag{57}$$

Next, we will show that there exists a PPT adversary $\mathcal{B}'$ such that $\epsilon' \leq \mathsf{Adv}^{\mathtt{PRF}}_{\mathsf{H},\mathcal{B}'}(\lambda) + \lambda(\lambda+1)\cdot 2^{-(\ell_{\mathsf{KEM}}+1)} + 2^{-\lambda}$. Now, we construct $\mathcal{B}'$ against the PRF-security of H as follows:

$\mathcal{B}'(1^\lambda)$**:** $\mathcal{B}'$ samples $\mathsf{c}_h \xleftarrow{\$} \{0,1\}^{\ell_{\mathsf{KEM}}}$ for every $h \in [\lambda+1]$, sends a query c_h to the function oracle for every $h \in [\lambda+1]$, and gets the answers $\mathsf{b}_1, \ldots, \mathsf{b}_{\lambda+1}$. If $\mathsf{b}_h \neq \mathsf{b}_{\lambda+1}$ for every $h \in [\lambda]$, then $\mathcal{B}'$ returns 1. Otherwise, $\mathcal{B}'$ returns 0.

The above completes the description of $\mathcal{B}'$. If $\mathcal{B}'$ gets outputs of a PRF H_k where $k \xleftarrow{\$} \mathcal{K}$ from the function oracle and $\mathcal{B}'$ returns 1, then its probability is just ϵ' by the definition, namely,

$$\Pr_{k \xleftarrow{\$} \mathcal{K}}[\mathcal{B}'^{\mathsf{H}_k(\cdot)}(1^\lambda) = 1] = \epsilon'.$$

For the case that $\mathcal{B}'$ gets outputs of a random function, we need to consider collisions among $\mathsf{c}_1, \ldots, \mathsf{c}_{\lambda+1}$. Let Coll be the event that there exist $i, j \in \{1, \ldots, \lambda+1\}$ with $i < j$ such that $\mathsf{c}_i = \mathsf{c}_j$. Then, by the union bound,

$$\Pr[\mathsf{Coll}] \leq \binom{\lambda+1}{2} \cdot 2^{-\ell} = \lambda(\lambda+1) \cdot 2^{-(\ell+1)}.$$

If Coll does not occur, all of $\mathsf{b}_1, \ldots, \mathsf{b}_\lambda$ are distributed uniformly at random, and hit the complement of $\mathsf{b}_{\lambda+1}$, which occurs with probability exactly $2^{-\lambda}$. Hence, we have

$$\Pr[\mathcal{B}'^{f(\cdot)}(1^\lambda) = 1 \mid \overline{\mathsf{Coll}}] = 2^{-\lambda}.$$

Therefore, it holds that

$$\begin{aligned}\Pr_{f \xleftarrow{\$} \mathsf{FUNC}_{\{0,1\}^{\ell_{\mathsf{KEM}}} \to \{0,1\}}}[\mathcal{B}'^{f(\cdot)}(1^\lambda) = 1] &\leq \Pr[\mathsf{Coll}] + \Pr[\mathcal{B}'^{f(\cdot)}(1^\lambda) = 1 \mid \overline{\mathsf{Coll}}] \\ &\leq \lambda(\lambda+1) \cdot 2^{-(\ell+1)} + 2^{-\lambda}.\end{aligned}$$

Then, we have

$$\begin{aligned}\mathsf{Adv}^{\mathsf{PRF}}_{\mathsf{H},\mathcal{B}'}(\lambda) &\geq \Pr_{k \xleftarrow{\$} \mathcal{K}}[\mathcal{B}'^{\mathsf{H}_k(\cdot)}(1^\lambda) = 1] - \Pr_{f \xleftarrow{\$} \mathsf{FUNC}_{\{0,1\}^\ell \to \{0,1\}}}[\mathcal{B}'^{f(\cdot)}(1^\lambda) = 1] \\ &\geq \epsilon' - \left(\lambda(\lambda+1) \cdot 2^{-(\ell_{\mathsf{KEM}}+1)} + 2^{-\lambda}\right),\end{aligned}$$

which implies

$$\epsilon' \leq \mathsf{Adv}^{\mathsf{PRF}}_{\mathsf{H},\mathcal{B}'}(\lambda) + \lambda(\lambda+1) \cdot 2^{-(\ell+1)} + 2^{-\lambda}. \tag{58}$$

By Eqs. (57) and (58), we can conclude that

$$\begin{aligned}\epsilon &\leq \left|\epsilon - \epsilon'\right| + \epsilon' \\ &\leq (\lambda+1)\mathsf{Adv}^{\mathtt{CTX\text{-}PR}}_{\mathsf{KEM},\mathcal{B}}(\lambda) + \mathsf{Adv}^{\mathsf{PRF}}_{\mathsf{H},\mathcal{B}'}(\lambda) + \lambda(\lambda+1) \cdot 2^{-(\ell_{\mathsf{KEM}}+1)} + 2^{-\lambda}.\end{aligned}$$

This completes the proof. □ (**Lemma 11**)

D.7 Proof of Lemma 12

The proof here is somewhat analogous to the proof for the leftover hash lemma [19]. For simplicity of notation, in this proof, we omit how the values appearing in the subscripts of the expectations and probabilities, which are sampled as in the distributions D_{real} and D_{rand} where $(\mathsf{c}'_{\mathsf{KEM}}, \mathsf{K}')$ denotes an independent copy of $(\mathsf{c}_{\mathsf{KEM}}, \mathsf{K})$ generated by $(\mathsf{c}'_{\mathsf{KEM}}, \mathsf{K}') \leftarrow \mathsf{Encap}(\mathsf{pk})$.

$$
\begin{aligned}
&\mathbf{SD}\big(D_{\mathsf{real}}, D_{\mathsf{rand}}\big) \\
&= \mathbf{E}_{\mathsf{pp}',k,(\mathsf{pk},\mathsf{sk})}\Big[\mathbf{SD}\big(\mathsf{H}_k(\mathsf{c}_{\mathsf{KEM}}), y\big)\Big] \\
&= \mathbf{E}_{\mathsf{pp}',k,(\mathsf{pk},\mathsf{sk})}\left[\frac{1}{2}\sum_{y'\in\{0,1\}}\Big|\Pr_{(\mathsf{c}_{\mathsf{KEM}},\mathsf{K})}\big[\mathsf{H}_k(\mathsf{c}_{\mathsf{KEM}}) = y'\big] - \Pr_{y}\big[y = y'\big]\Big|\right] \\
&= \frac{1}{2}\mathbf{E}_{\mathsf{pp}',k,(\mathsf{pk},\mathsf{sk})}\left[\sum_{y'\in\{0,1\}}\Big|\Pr_{(\mathsf{c}_{\mathsf{KEM}},\mathsf{K})}\big[\mathsf{H}_k(\mathsf{c}_{\mathsf{KEM}}) = y'\big] - \frac{1}{2}\Big|\right] \\
&\le \frac{1}{2}\mathbf{E}_{\mathsf{pp}',k,(\mathsf{pk},\mathsf{sk})}\left[\left(\sum_{y'\in\{0,1\}}\Big(\Pr_{(\mathsf{c}_{\mathsf{KEM}},\mathsf{K})}\big[\mathsf{H}_k(\mathsf{c}_{\mathsf{KEM}}) = y'\big] - \frac{1}{2}\Big)^2\right)^{\frac{1}{2}} \cdot \Big(\sum_{y'\in\{0,1\}} 1^2\Big)^{\frac{1}{2}}\right] \qquad (59) \\
&= \frac{1}{\sqrt{2}}\mathbf{E}_{\mathsf{pp}',k,(\mathsf{pk},\mathsf{sk})}\left[\left(\sum_{y'\in\{0,1\}}\Pr_{(\mathsf{c}_{\mathsf{KEM}},\mathsf{K})}\big[\mathsf{H}_k(\mathsf{c}_{\mathsf{KEM}}) = y'\big]^2\right.\right. \\
&\qquad\qquad \left.\left. - \sum_{y'\in\{0,1\}}\Pr_{(\mathsf{c}_{\mathsf{KEM}},\mathsf{K})}\big[\mathsf{H}_k(\mathsf{c}_{\mathsf{KEM}}) = y'\big] + \sum_{y'\in\{0,1\}}\Big(\frac{1}{4}\Big)\right)^{\frac{1}{2}}\right] \\
&= \frac{1}{\sqrt{2}}\mathbf{E}_{\mathsf{pp}',k,(\mathsf{pk},\mathsf{sk})}\left[\Big(\sum_{y'\in\{0,1\}}\Pr_{(\mathsf{c}_{\mathsf{KEM}},\mathsf{K})}\big[\mathsf{H}_k(\mathsf{c}_{\mathsf{KEM}}) = y'\big]^2 - 1 + \frac{2}{4}\Big)^{\frac{1}{2}}\right] \\
&= \frac{1}{\sqrt{2}}\mathbf{E}_{\mathsf{pp}',k,(\mathsf{pk},\mathsf{sk})}\left[\Big(\Pr_{(\mathsf{c}_{\mathsf{KEM}},\mathsf{K}),(\mathsf{c}'_{\mathsf{KEM}},\mathsf{K}')}\big[\mathsf{H}_k(\mathsf{c}_{\mathsf{KEM}}) = \mathsf{H}_k(\mathsf{c}'_{\mathsf{KEM}})\big] - \frac{1}{2}\Big)^{\frac{1}{2}}\right] \qquad (60) \\
&\le \frac{1}{\sqrt{2}}\left(\mathbf{E}_{\mathsf{pp}',k,(\mathsf{pk},\mathsf{sk})}\Big[\Pr_{(\mathsf{c}_{\mathsf{KEM}},\mathsf{K}),(\mathsf{c}'_{\mathsf{KEM}},\mathsf{K}')}\big[\mathsf{H}_k(\mathsf{c}_{\mathsf{KEM}}) = \mathsf{H}_k(\mathsf{c}'_{\mathsf{KEM}})\big]\Big] - \frac{1}{2}\right)^{\frac{1}{2}} \qquad (61) \\
&= \frac{1}{\sqrt{2}}\left(\Pr_{\mathsf{pp}',k,(\mathsf{pk},\mathsf{sk}),(\mathsf{c}_{\mathsf{KEM}},\mathsf{K}),(\mathsf{c}'_{\mathsf{KEM}},\mathsf{K}')}\Big[\mathsf{H}_k(\mathsf{c}_{\mathsf{KEM}}) = \mathsf{H}_k(\mathsf{c}'_{\mathsf{KEM}})\Big] - \frac{1}{2}\right)^{\frac{1}{2}}, \qquad (62)
\end{aligned}
$$

where Eq. (59) holds by the Cauchy-Schwarz inequality, Eq. (60) holds due to the definition of collision probability, and Eq. (61) holds by Jensen's inequality.

We will show that there exist PPT adversaries $\mathcal{A}$ attacking the PRF-security of H and $\mathcal{A}'$ attacking the ciphertext pseudorandomness of KEM such that

$$\begin{aligned} Q &:= \Pr_{\mathsf{pp}',k,(\mathsf{pk},\mathsf{sk}),(\mathsf{c}_{\mathsf{KEM}},\mathsf{K}),(\mathsf{c}'_{\mathsf{KEM}},\mathsf{K}')}\Big[\mathsf{H}_k(\mathsf{c}_{\mathsf{KEM}}) = \mathsf{H}_k(\mathsf{c}'_{\mathsf{KEM}})\Big] \\ &\leq \frac{1}{2} + \mathsf{Adv}^{\mathtt{PRF}}_{\mathsf{H},\mathcal{A}}(\lambda) + \mathsf{Adv}^{\mathtt{CTX\text{-}PR}}_{\mathsf{KEM},\mathcal{A}'}(\lambda) + 2^{-\ell_{\mathsf{KEM}}}. \end{aligned}$$

We construct the adversary $\mathcal{A}$ attacking the security of PRF as follows:

$\mathcal{A}^{f(\cdot)}(1^\lambda)$**:** $\mathcal{A}$ computes $\mathsf{pp}' \xleftarrow{\$} \mathsf{KSetup}(1^\lambda), (\mathsf{pk},\mathsf{sk}) \xleftarrow{\$} \mathsf{KKG}(\mathsf{pp}), (\mathsf{c}_{\mathsf{KEM}},\mathsf{K}) \xleftarrow{\$} \mathsf{Encap}(\mathsf{pk})$, and $(\mathsf{c}'_{\mathsf{KEM}},\mathsf{K}') \xleftarrow{\$} \mathsf{Encap}(\mathsf{pk})$. Then, $\mathcal{A}$ accesses its oracle with the input $\mathsf{c}_{\mathsf{KEM}}$ (resp. $\mathsf{c}'_{\mathsf{KEM}}$), and receives its output y (resp. y'). If $y = y'$, then $\mathcal{A}$ returns 1, otherwise, $\mathcal{A}$ returns 0.

From the design of $\mathcal{A}$, the following equalities hold:

$$\mathsf{Adv}^{\mathtt{PRF}}_{\mathsf{H},\mathcal{A}}(\lambda) = \Big|\Pr_{k\xleftarrow{\$}\mathcal{K}}\big[\mathcal{A}^{\mathsf{H}_k(\cdot)} = 1\big] - \Pr_{f\xleftarrow{\$}\mathsf{FUNC}_{\{0,1\}^{\ell_{\mathsf{KEM}}}\to\{0,1\}}}\big[\mathcal{A}^{f(\cdot)} = 1\big]\Big|,$$

$$\Pr_{k\xleftarrow{\$}\mathcal{K}}\big[\mathcal{A}^{\mathsf{H}_k(\cdot)} = 1\big] = \Pr_{\mathsf{pp}',k,(\mathsf{pk},\mathsf{sk}),(\mathsf{c}_{\mathsf{KEM}},\mathsf{K}),(\mathsf{c}'_{\mathsf{KEM}},\mathsf{K}')}\big[\mathsf{H}_k(\mathsf{c}_{\mathsf{KEM}}) = \mathsf{H}_k(\mathsf{c}'_{\mathsf{KEM}})\big] = Q,$$

$$\begin{aligned} &\Pr_{f\xleftarrow{\$}\mathsf{FUNC}_{\{0,1\}^{\ell_{\mathsf{KEM}}}\to\{0,1\}}}\big[\mathcal{A}^{f(\cdot)} = 1\big] \\ &= \Pr_{f,\mathsf{pp},(\mathsf{pk},\mathsf{sk}),(\mathsf{c}_{\mathsf{KEM}},\mathsf{K}),(\mathsf{c}'_{\mathsf{KEM}},\mathsf{K}')}\big[f(\mathsf{c}_{\mathsf{KEM}}) = f(\mathsf{c}'_{\mathsf{KEM}})\big] \\ &= \Pr_{f,\mathsf{pp},(\mathsf{pk},\mathsf{sk}),(\mathsf{c}_{\mathsf{KEM}},\mathsf{K}),(\mathsf{c}'_{\mathsf{KEM}},\mathsf{K}')}\big[f(\mathsf{c}_{\mathsf{KEM}}) = f(\mathsf{c}'_{\mathsf{KEM}}) \wedge \mathsf{c}_{\mathsf{KEM}} \neq \mathsf{c}'_{\mathsf{KEM}}\big] \\ &\qquad + \Pr_{\mathsf{pp},(\mathsf{pk},\mathsf{sk}),(\mathsf{c}_{\mathsf{KEM}},\mathsf{K}),(\mathsf{c}'_{\mathsf{KEM}},\mathsf{K}')}\big[\mathsf{c}_{\mathsf{KEM}} = \mathsf{c}'_{\mathsf{KEM}}\big] \\ &= \frac{1}{2} + \Pr_{\mathsf{pp},(\mathsf{pk},\mathsf{sk}),(\mathsf{c}_{\mathsf{KEM}},\mathsf{K}),(\mathsf{c}'_{\mathsf{KEM}},\mathsf{K}')}\big[\mathsf{c}_{\mathsf{KEM}} = \mathsf{c}'_{\mathsf{KEM}}\big] \end{aligned}$$

Therefore,

$$\mathsf{Adv}^{\mathtt{PRF}}_{\mathsf{H},\mathcal{A}}(\lambda) = \Big|Q - \big(\frac{1}{2} + \Pr_{\mathsf{pp},(\mathsf{pk},\mathsf{sk}),(\mathsf{c}_{\mathsf{KEM}},\mathsf{K}),(\mathsf{c}'_{\mathsf{KEM}},\mathsf{K}')}\big[\mathsf{c}_{\mathsf{KEM}} = \mathsf{c}'_{\mathsf{KEM}}\big]\big)\Big| \qquad (63)$$

We construct the adversary $\mathcal{A}'$ attacking the ciphertext pseudorandomness of KEM as follows:

$\mathcal{A}'(\mathsf{pp},\mathsf{pk},\mathsf{c}_{\mathsf{KEM}})$**:** $\mathcal{A}'$ computes $(\mathsf{c}'_{\mathsf{KEM}},\mathsf{K}') \xleftarrow{\$} \mathsf{Encap}(\mathsf{pk})$. If $\mathsf{c}_{\mathsf{KEM}} = \mathsf{c}'_{\mathsf{KEM}}$, then $\mathcal{A}'$ returns 1, otherwise, $\mathcal{A}$ returns 0.

From the design of $\mathcal{A}'$, the following equalities hold:

$$\mathsf{Adv}^{\mathtt{CTX\text{-}PR}}_{\mathsf{KEM},\mathcal{A}'}(\lambda)$$

$$
\begin{aligned}
&= \Bigg| \Pr_{\mathsf{pp},(\mathsf{pk},\mathsf{sk})}\left[(\mathsf{c}_{\mathsf{KEM}},\mathsf{K}) \xleftarrow{\$} \mathsf{Encap}(\mathsf{pk}) : \mathcal{A}'(\mathsf{pp},\mathsf{pk},\mathsf{c}_{\mathsf{KEM}}) = 1\right] \\
&\qquad\qquad - \Pr_{\mathsf{pp},(\mathsf{pk},\mathsf{sk})}\left[\mathsf{c}_{\mathsf{KEM}} \xleftarrow{\$} \{0,1\}^{\ell_{\mathsf{KEM}}} : \mathcal{A}'(\mathsf{pp},\mathsf{pk},\mathsf{c}_{\mathsf{KEM}}) = 1\right]\Bigg| \\
&= \Bigg| \Pr_{\mathsf{pp},(\mathsf{pk},\mathsf{sk})}\left[(\mathsf{c}_{\mathsf{KEM}},\mathsf{K}),(\mathsf{c}'_{\mathsf{KEM}},\mathsf{K}') \xleftarrow{\$} \mathsf{Encap}(\mathsf{pk}) : \mathsf{c}_{\mathsf{KEM}} = \mathsf{c}'_{\mathsf{KEM}}\right] \\
&\qquad\qquad - \Pr_{\mathsf{pp},(\mathsf{pk},\mathsf{sk})}\left[\begin{array}{l}\mathsf{c}_{\mathsf{KEM}} \xleftarrow{\$} \{0,1\}^{\ell_{\mathsf{KEM}}}, \\ (\mathsf{c}'_{\mathsf{KEM}},\mathsf{K}') \xleftarrow{\$} \mathsf{Encap}(\mathsf{pk})\end{array} : \mathsf{c}_{\mathsf{KEM}} = \mathsf{c}'_{\mathsf{KEM}}\right]\Bigg| \\
&= \Bigg| \Pr_{\mathsf{pp},(\mathsf{pk},\mathsf{sk})}\left[(\mathsf{c}_{\mathsf{KEM}},\mathsf{K}),(\mathsf{c}'_{\mathsf{KEM}},\mathsf{K}') \xleftarrow{\$} \mathsf{Encap}(\mathsf{pk}) : \mathsf{c}_{\mathsf{KEM}} = \mathsf{c}'_{\mathsf{KEM}}\right] - 2^{-\ell_{\mathsf{KEM}}}\Bigg|.
\end{aligned}
$$

Therefore, by the triangle inequality, the following holds:

$$
\begin{aligned}
&\Pr_{\mathsf{pp},(\mathsf{pk},\mathsf{sk})}\left[(\mathsf{c}_{\mathsf{KEM}},\mathsf{K}),(\mathsf{c}'_{\mathsf{KEM}},\mathsf{K}') \xleftarrow{\$} \mathsf{Encap}(\mathsf{pk}) : \mathsf{c}_{\mathsf{KEM}} = \mathsf{c}'_{\mathsf{KEM}}\right] \\
&\leq \mathsf{Adv}^{\mathtt{CTX\text{-}PR}}_{\mathsf{KEM},\mathcal{A}'}(\lambda) + 2^{-\ell_{\mathsf{KEM}}}. \qquad (64)
\end{aligned}
$$

From Eqs. (63) and (64), the following holds:

$$
Q \leq \frac{1}{2} + \mathsf{Adv}^{\mathtt{PRF}}_{\mathsf{H},\mathcal{A}}(\lambda) + \mathsf{Adv}^{\mathtt{CTX\text{-}PR}}_{\mathsf{KEM},\mathcal{A}'}(\lambda) + 2^{-\ell_{\mathsf{KEM}}}. \qquad (65)
$$

From Eqs. (62) and (65), we finally have that there exist PPT adversaries $\mathcal{A}$ and $\mathcal{A}'$ such that

$$
\begin{aligned}
&\mathbf{SD}\left(D_{\mathsf{real}}, D_{\mathsf{rand}}\right) \\
&\leq \frac{1}{\sqrt{2}}\left(\frac{1}{2} + \mathsf{Adv}^{\mathtt{PRF}}_{\mathsf{H},\mathcal{A}}(\lambda) + \mathsf{Adv}^{\mathtt{CTX\text{-}PR}}_{\mathsf{KEM},\mathcal{A}'}(\lambda) + 2^{-\ell_{\mathsf{KEM}}} - \frac{1}{2}\right)^{\frac{1}{2}} \\
&= \frac{1}{\sqrt{2}}\left(\mathsf{Adv}^{\mathtt{PRF}}_{\mathsf{H},\mathcal{A}}(\lambda) + \mathsf{Adv}^{\mathtt{CTX\text{-}PR}}_{\mathsf{KEM},\mathcal{A}'}(\lambda) + 2^{-\ell_{\mathsf{KEM}}}\right)^{\frac{1}{2}}.
\end{aligned}
$$

□ (**Lemma 12**)

D.8 Proof of Lemma 13

Using $\mathcal{A}$ as a building block, we construct a reduction algorithm $\mathcal{B}$ attacking the pseudorandomness of KEM.

$\mathcal{B}(\mathsf{pp}',\mathsf{pk},\mathsf{c}_{\mathsf{KEM}},\mathsf{K})$**:** $\mathcal{B}$ picks $k \in \mathcal{K}$ and $t \in [q]$ uniformly at random, sets $\mathsf{dpk} \leftarrow \mathsf{pk}$ and $\mathsf{pp} \leftarrow (\mathsf{pp}',k)$, and computes $(\mathsf{fpk}_i,\mathsf{fsk}_i) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp}')$ for each $i \in [\ell]$. $\mathcal{B}$ runs $\mathcal{A}(\mathsf{pp},(\mathsf{fpk}_i,\mathsf{fsk}_i)_{i\in[\ell]},\mathsf{dpk})$ and executes the following procedure for answering $\mathcal{A}$'s s-th query $((\mathsf{fm}_i)_{i\in[\ell]},\mathsf{dm})$ to the randomness oracle where $s \in [q]$:

– If $s < t$, then $\mathcal{B}$ computes $(r_i)_{i\in[\ell]}$ in the same way as $\mathcal{O}^{\mathtt{Rand}}_{\mathsf{Game1}}$, and returns $(r_i)_{i\in[\ell]}$ to $\mathcal{A}$.
– If $s = t$, then $\mathcal{B}$ executes the following procedure and returns its result $(r_i)_{i\in[\ell]}$ to $\mathcal{A}$ as its answer to $\mathcal{A}$'s t-th query.

$\mathsf{c}_{\mathsf{DEM}} \leftarrow \mathsf{DEnc}(\mathsf{K}, \mathsf{dm})$
$\mathtt{b}_1 \| \mathtt{b}_2 \| \cdots \| \mathtt{b}_\ell := (\mathsf{c}_{\mathsf{KEM}}, \mathsf{c}_{\mathsf{DEM}})$
for $i \in [\ell]$
  $\mathtt{ctr} \leftarrow 0$
  do
    $r^* \xleftarrow{\$} \mathcal{R}$
    $(\mathsf{c}^*_{\mathsf{KEM}}, \mathsf{K}^*) \leftarrow \mathsf{Encap}(\mathsf{fpk}_i; r^*)$
    $\mathtt{b}^* \leftarrow \mathsf{H}_k(\mathsf{c}^*_{\mathsf{KEM}})$
    $\mathtt{ctr} \leftarrow \mathtt{ctr} + 1$
  until $\mathtt{b}^* = \mathtt{b}_i \vee \mathtt{ctr} \geq \lambda$
  $r_i \leftarrow r^*$

– If $s > t$, then $\mathcal{B}$ computes $(r_i)_{i\in[\ell]}$ in the same way as $\mathcal{O}^{\mathtt{Rand}}_{\mathsf{Game0}}$, and returns $(r_i)_{i\in[\ell]}$ to $\mathcal{A}$.

Finally, $\mathcal{A}$ returns a bit b' and $\mathcal{B}$ returns b'.

The above completes the description of $\mathcal{B}$. Let b be $\mathcal{B}$'s challenge bit. It is not hard to see the following facts:

– If $b = 0$ and $t = 1$, then $\mathcal{B}$ simulates Game 0 perfectly for $\mathcal{A}$. Therefore, the probability that $\mathcal{B}$ returns $b' = 1$ is identical to the probability that $\mathcal{A}$ returns 1 in Game 0, namely,

$$\Pr[b' = 1 \mid b = 0 \wedge t = 1] = \Pr[\mathsf{G}_0]. \tag{66}$$

– If $b = 1$ and $t = q$, then $\mathcal{B}$ simulates Game 1 perfectly for $\mathcal{A}$. Therefore, the probability that $\mathcal{B}$ returns $b' = 1$ is identical to the probability that $\mathcal{A}$ returns 1 in Game 1, namely,

$$\Pr[b' = 1 \mid b = 1 \wedge t = q] = \Pr[\mathsf{G}_1]. \tag{67}$$

– From $\mathcal{A}$'s view point, for any $s \in [q-1]$, the experiment simulated by $\mathcal{B}$ in the following situations is identical:
  • $b = 1$ and $t = s$.
  • $b = 0$ and $t = s + 1$.

  Since $\mathcal{B}$ outputs $\mathcal{A}$'s output as it is, we have

$$\Pr[b' = 1 \mid b = 1 \wedge t = s] = \Pr[b' = 1 \mid b = 0 \wedge t = s + 1]. \tag{68}$$

Hence, we can compute $\mathcal{B}$'s advantage against the pseudorandomness of KEM as follows:

$$\mathsf{Adv}^{\mathtt{PR}}_{\mathsf{KEM},\mathcal{B}}(\lambda) = \Big|\Pr[b' = 1 \mid b = 0] - \Pr[b' = 1 \mid b = 1]\Big|$$

$$= \Big| \sum_{s\in[q]} \Pr[j=s] \Big(\Pr[b'=1 \mid b=0 \wedge t=s] - \Pr[b'=1 \mid b=1 \wedge t=s] \Big) \Big|$$

$$= \frac{1}{q} \Big| \Pr[b'=1 \mid b=0 \wedge t=1] - \Pr[b'=1 \mid b=1 \wedge t=q] \Big| \quad (69)$$

$$= \frac{1}{q} \Big| \Pr[\mathsf{G}_0] - \Pr[\mathsf{G}_1] \Big|, \quad (70)$$

where Eq. (69) follows from Eq. (68), and Eq. (70) follows from Eqs. (66) and (67). This completes the proof. □ (**Lemma 13**)

D.9 Proof of Lemma 14

Using $\mathcal{A}$ as a building block, we construct a reduction algorithm $\mathcal{B}' = (\mathcal{B}'_0, \mathcal{B}'_1)$ attacking the pseudorandomness of DEM.

$\mathcal{B}'_0(1^\lambda)$**:** $\mathcal{B}'_0$ picks $k \in \mathcal{K}$ and $t \in [q]$ uniformly at random, computes $\mathsf{pp}' \xleftarrow{\$} \mathsf{KSetup}(1^\lambda)$, sets $\mathsf{pp} \leftarrow (\mathsf{pp}', k)$, and computes $(\mathsf{dpk}, \mathsf{dsk}) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp})$ and $(\mathsf{fpk}_i, \mathsf{fsk}_i) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp})$ for each $i \in [\ell]$. $\mathcal{B}'_0$ runs $\mathcal{A}(\mathsf{pp}, (\mathsf{fpk}_i, \mathsf{fsk}_i)_{i\in[\ell]}, \mathsf{dpk})$ and executes the following procedure for answering $\mathcal{A}$'s s-th query $((\mathsf{fm}_i)_{i\in[\ell]}, \mathsf{dm})$ to the randomness oracle where $s \in [t]$:

- If $s < t$, then $\mathcal{B}'_0$ computes $(r_i)_{i\in[\ell]}$ in the same way as $\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game2}}$, and returns $(r_i)_{i\in[\ell]}$ to $\mathcal{A}$.
- If $s = t$, then $\mathcal{B}'_0$ sets st as all the values known to $\mathcal{B}'_0$, and terminates with output $(\mathsf{dm}, \mathrm{st})$.

$\mathcal{B}'_1(\mathsf{c}_{\mathsf{DEM}}, \mathrm{st})$**:** $\mathcal{B}'_1$ executes the following procedure and returns its result $(r_i)_{i\in[\ell]}$ to $\mathcal{A}$ as its answer to $\mathcal{A}$'s t-th query.

$\mathsf{c}_{\mathsf{KEM}} \xleftarrow{\$} \{0,1\}^{\ell_{\mathsf{KEM}}}$
$\mathsf{b}_1 \| \mathsf{b}_2 \| \cdots \| \mathsf{b}_\ell := (\mathsf{c}_{\mathsf{KEM}}, \mathsf{c}_{\mathsf{DEM}})$
for $i \in [\ell]$
  $\mathtt{ctr} \leftarrow 0$
  do
    $r^* \xleftarrow{\$} \mathcal{R}$
    $(\mathsf{c}^*_{\mathsf{KEM}}, \mathsf{K}^*) \leftarrow \mathsf{Encap}(\mathsf{fpk}_i; r^*)$
    $\mathsf{b}^* \leftarrow \mathsf{H}_k(\mathsf{c}^*_{\mathsf{KEM}})$
    $\mathtt{ctr} \leftarrow \mathtt{ctr} + 1$
  until $\mathsf{b}^* = \mathsf{b}_i \vee \mathtt{ctr} \geq \lambda$
  $r_i \leftarrow r^*$
return $(r_i)_{i\in[\ell]}$

For the s-th query to the randomness oracle from $\mathcal{A}$ where $s > t$, $\mathcal{B}'_1$ computes $(r_i)_{i\in[\ell]}$ in the same way as $\mathcal{O}^{\mathsf{Rand}}_{\mathsf{Game1}}$, and returns $(r_i)_{i\in[\ell]}$ to $\mathcal{A}$. Finally, $\mathcal{A}$ returns a bit b' and $\mathcal{B}'_1$ returns b'.

The above completes the description of $\mathcal{B}'$. Let b be $\mathcal{B}'$'s challenge bit. It is not hard to see the following facts:

– If $b = 0$ and $t = 1$, then $\mathcal{B}'$ simulates Game 1 perfectly for $\mathcal{A}$. Therefore, the probability that $\mathcal{B}'$ returns $b' = 1$ is identical to the probability that $\mathcal{A}$ returns 1 in Game 1, namely,

$$\Pr[b' = 1 \mid b = 0 \wedge t = 1] = \Pr[\mathsf{G}_1]. \tag{71}$$

– If $b = 1$ and $t = q$, then $\mathcal{B}'$ simulates Game 2 perfectly for $\mathcal{A}$. Therefore, the probability that $\mathcal{B}'$ returns $b' = 1$ is identical to the probability that $\mathcal{A}$ returns 1 in Game 2, namely,

$$\Pr[b' = 1 \mid b = 1 \wedge t = q] = \Pr[\mathsf{G}_2]. \tag{72}$$

– From $\mathcal{A}$'s view point, for any $s \in [q-1]$, the experiment simulated by $\mathcal{B}'$ in the following situations is identical:
  - $b = 1$ and $t = s$.
  - $b = 0$ and $t = s + 1$.

  Since $\mathcal{B}'$ outputs $\mathcal{A}$'s output as it is, we have

$$\Pr[b' = 1 \mid b = 1 \wedge t = s] = \Pr[b' = 1 \mid b = 0 \wedge t = s + 1]. \tag{73}$$

Hence, we can compute $\mathcal{B}'$'s advantage against the pseudorandomness of DEM as follows:

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{PR}}_{\mathsf{DEM},\mathcal{B}'}(\lambda) &= \Big|\Pr[b' = 1 \mid b = 0] - \Pr[b' = 1 \mid b = 1]\Big| \\
&= \Big|\sum_{s\in[q]} \Pr[t = s]\Big(\Pr[b' = 1 \mid b = 0 \wedge t = s] \\
&\qquad\qquad - \Pr[b' = 1 \mid b = 1 \wedge t = s]\Big)\Big| \\
&= \frac{1}{q}\Big|\Pr[b' = 1 \mid b = 0 \wedge t = 1] - \Pr[b' = 1 \mid b = 1 \wedge t = q]\Big| \qquad (74) \\
&= \frac{1}{q}\Big|\Pr[\mathsf{G}_1] - \Pr[\mathsf{G}_2]\Big|, \qquad (75)
\end{aligned}$$

where Eq. (74) follows from Eq. (73), and Eq. (75) follows from Eqs. (71) and (72). This completes the proof. □ (**Lemma 14**)

E Special PRF Property of Hashed ElGamal KEM

The hashed ElGamal KEM, when described as a module-level KEM, is constructed as in Fig. 16. It does not use KEMTk, KEMTg, and KEMVf, and thus these algorithms are omitted from the description.

Theorem 15. *The hashed ElGamal KEM satisfies the special PRF property in the random oracle model where H is modeled as a random oracle.*

$\mathsf{KSetup}(1^\lambda)$:	$\mathsf{KEMEk}(\mathsf{pp})$:	$\mathsf{KEMKg}(\mathsf{ek}, r \in \mathbb{Z}_p^*)$:
$g \xleftarrow{\$} \mathbb{G}$	$\mathsf{dk} \xleftarrow{\$} \mathbb{Z}_p^*$	return $\mathsf{K} \leftarrow H(\mathsf{ek}^r)$
Let $H : \mathbb{G} \to \{0,1\}^\ell$	$\mathsf{ek} \leftarrow g^{\mathsf{dk}}$	$\mathsf{KEMCg}(r \in \mathbb{Z}_p^*)$:
return $\mathsf{pp} \leftarrow (g, H)$	return $(\mathsf{ek}, \mathsf{dk})$	return $\mathsf{c} \leftarrow g^r$
		$\mathsf{KEMKd}(\mathsf{dk}, \mathsf{c})$:
		return $\mathsf{K} \leftarrow H(\mathsf{c}^{\mathsf{dk}})$

Fig. 16. The hashed ElGamal KEM described as a module-level KEM. This module-level KEM does not use KEMTk, KEMTg, and KEMVf, and thus these algorithms are omitted. In the figure, $\mathbb{G}$ is a cyclic group with prime order $p = \Omega(2^\lambda)$.

Proof. Let $\mathcal{A}$ be an adversary that attacks the special PRF property of the hashed ElGamal KEM, and makes q_H random oracle queries and q_f function queries. Without loss of generality, we assume that $\mathcal{A}$ does not repeat the same query twice.

Let $S_H = \{x_i\}_{i \in [q_H]} \subseteq \mathbb{G}$ be the set of the random oracle queries made by $\mathcal{A}$, where x_i denotes $\mathcal{A}$'s i-th query to the random oracle H. Let $S_f = \{r_j\}_{j \in [q_f]} \subseteq \mathbb{Z}_p^*$ be the set of the function queries made by $\mathcal{A}$, where r_j denotes $\mathcal{A}$'s j-th query to the function oracle (i.e. either the real function $\mathsf{KEMKg}(\mathsf{ek}, \cdot)$ or a random function $f(\cdot)$). Furthermore, let $S'_f := \{\mathsf{ek}^r | r \in S_f\}$.

Observe that during the experiment for the special PRF property, the only elements that are input to the random oracle H are those in $S_H \cup S'_f$. Hence, if there is no overlap between S_H and S'_f, then $\mathcal{A}$ cannot distinguish whether it has access to the real function $\mathsf{KEMKg}(\mathsf{ek}, \cdot)$ or a random function $f(\cdot)$ chosen as $f \xleftarrow{\$} \mathsf{FUNC}_{\mathbb{Z}_p^* \to \{0,1\}^\ell}$. Hence, $\mathcal{A}$'s distinguishing advantage is bounded by the probability that S_H and S'_f have an overlap. Since the information of $\mathsf{ek} = g^{\mathsf{dk}}$ is never leaked in the experment for the special PRF property, and $\mathsf{dk} \in \mathbb{Z}_p^*$ is chosen uniformly at random, the probability that S_H and S'_f have an overlap can be calculated as

$$\begin{aligned}
\Pr_{\mathsf{dk} \xleftarrow{\$} \mathbb{Z}_p^*}[S'_f \cap S_H \neq \emptyset] &= \Pr_{\mathsf{dk} \xleftarrow{\$} \mathbb{Z}_p^*}[\{g^{\mathsf{dk} \cdot r_j}\}_{j \in [q_f]} \cap \{x_i\}_{i \in [q_H]} \neq \emptyset] \\
&\leq \sum_{j \in [q_f]} \Pr_{\mathsf{dk} \xleftarrow{\$} \mathbb{Z}_p^*}[g^{\mathsf{dk} \cdot r_j} \in \{x_i\}_{i \in [q_H]}] \\
&= \sum_{j \in [q_f]} \Pr_{\mathsf{dk} \xleftarrow{\$} \mathbb{Z}_p^*}[g^{\mathsf{dk}} \in \{(x_i)^{1/r_j}\}_{i \in [q_H]}] \\
&\leq \sum_{j \in [q_f]} \frac{q_H}{p-1} = \frac{q_f q_H}{p-1},
\end{aligned}$$

which is negligible in λ. Hence, the hashed ElGamal KEM satisfies the special PRF property in the random oracle model. □ (**Theorem 15**)