CPA-secure KEMs are also sufficient for Post-Quantum TLS 1.3

Biming Zhou[1,3], Haodong Jiang[2], and Yunlei Zhao[1,3]

[1] College of Computer Science and Technology, Fudan University, Shanghai 200433, China
[2] Henan Key Laboratory of Network Cryptography Technology, Zhengzhou, 450001, Henan, China
[3] State Key Laboratory of Cryptology, Beijing 100878, China
bmzhou22@m.fudan.edu.cn, hdjiang13@163.com, ylzhao@fudan.edu.cn

Abstract. In the post-quantum migration of TLS 1.3, an ephemeral Diffie-Hellman must be replaced with a post-quantum key encapsulation mechanism (KEM). At EUROCRYPT 2022, Huguenin-Dumittan and Vaudenay [22] demonstrated that KEMs with standard CPA security are sufficient for the security of the TLS 1.3 handshake. However, their result is only proven in the random oracle model (ROM), and as the authors comment, their reduction is very much non-tight and not sufficient to guarantee security in practice due to the $O(q^6)$-loss, where q is the number of adversary's queries to random oracles. Moreover, in order to analyze the post-quantum security of TLS 1.3 handshake with a KEM, it is necessary to consider the security in the quantum ROM (QROM). Therefore, they leave the tightness improvement of their ROM proof and the QROM proof of such a result as an interesting open question.
In this paper, we resolve this problem. We improve the ROM proof in [22] from an $O(q^6)$-loss to an $O(q)$-loss with standard CPA-secure KEMs which can be directly obtained from the underlying public-key encryption (PKE) scheme in CRYSTALS-Kyber [10]. Moreover, we show that if the KEMs are constructed from rigid deterministic public-key encryption (PKE) schemes such as the ones in Classic McEliece [2] and NTRU [13], this $O(q)$-loss can be further improved to an $O(1)$-loss. Hence, our reductions are sufficient to guarantee security in practice. According to our results, a CPA-secure KEM (which is more concise and efficient than the currently used CCA/1CCA-secure KEM) can be directly employed to construct a post-quantum TLS 1.3. Furthermore, we lift our ROM result into QROM and first prove that the CPA-secure KEMs are also sufficient for the post-quantum TLS 1.3 handshake. In particular, the techniques introduced to improve reduction tightness in this paper may be of independent interest.

Keywords: TLS 1.3 · tightness · quantum random oracle model · KEM-TLS

Table of Contents

1 Introduction

The Transport Layer Security (TLS) protocol is one of the most widely deployed cryptographic protocols in practice. As the NIST standardization of post-quantum cryptography (PQC) progresses, exploring the transition of the TLS 1.3 protocol to post-quantum (PQ) security has become a significant topic. To ensure PQ security for parts of the protocol that use Diffie-Hellman (DH) key exchange, it is necessary to replace the existing DH key exchange with a PQ secure key encapsulation mechanism (KEM).

The existing TLS 1.3 protocol [17], as well as the majority of other cryptographic protocols such as KEM-TLS [34], Signal [11], and Noise [4] schemes, rely on the PRF-ODH [12] assumption to achieve security. The PRF-ODH assumption is a variant of the hashed DH assumption. For PQ variants of these protocols, it has been shown that IND-1CCA security is required for the replaced KEMs, see PQ TLS [17, 22, 34, 35], PQ Signal [11], and PQ Noise [4]. Simply put, IND-1CCA security ensures that any probabilistic polynomial time (PPT) adversary cannot distinguish between a legitimately generated key and a random key with at most one decapsulation query. Usually, IND-CCA secure KEMs are taken as IND-1CCA secure KEMs for the implementation of PQ TLS 1.3 [1]. IND-CCA security is typically achieved by employing a Fujisaki-Okamoto-like (FO-like) transformation to an OW-CPA/IND-CPA secure public-key encryption (PKE), see [8, 16, 18–21, 25–28]. In particular, CRYSTALS-Kyber [10] and the remaining KEMs in the Round-4 submissions [30] all employ an FO-like transformation. However, the FO-like transformation in IND-CCA secure KEMs requires re-encryption in decapsulation, significantly impacting decapsulation efficiency as demonstrated in [22]. For instance, there is a 2.17X speedup over decapsulation in CRYSTALS-Kyber [10], and a 6.11X speedup in FrodoKEM [29] when re-encryption is removed, as shown in [22]. Moreover, re-encryption can render KEMs more vulnerable to side-channel attacks as demonstrated in [5, 37], affecting nearly all NIST-PQC Round-3 KEMs. Therefore, the design of an IND-1CCA secure KEM without re-encryption was left as an open problem in [34].

Huguenin-Dumittan and Vaudenay [22] made the first attempt to solve this problem by proposing two general constructions of IND-1CCA secure KEMs from OW-CPA/IND-CPA PKEs. One construction, denoted as T_{CH}, incorporates a key-confirmation component into the original ciphertext, causing ciphertext expansion. Another construction, denoted as T_H, works without ciphertext expansion, and the key is derived by $H(m, c)$. Building upon [22], Jiang et al. [24] introduced an implicit variant of T_H, denoted as T_{RH}, and provided a tighter security reduction for both T_H and T_{RH} in the Random Oracle Model (ROM) compared to the proof presented in [22]. Jiang et al. [24] also established the security of T_H and T_{RH} in the Quantum Random Oracle Model (QROM) by introducing a variant of the measure-and-reprogram technique [14, 15].

Paquin, Stebila, and Tamvada [33] conjectured that CPA KEMs are sufficient for TLS 1.3. As shown in Fig. 1, the construction of CPA KEMs is more concise than 1CCA KEMs as one hash calculation can be removed. Huguenin-Dumittan and Vaudenay [22] confirmed this conjecture. They observed that in the TLS

1.3 key schedule, the keys are obtained by applying key-derivation functions (KDFs) to the shared secret and the hash of the transcript so far (including the ciphertext). Inspired by the proof of security of the T_H transform, they proved that if the underlying KEM is OW-CPA secure, then the TLS 1.3 handshake protocol is secure in the MultiStage model of Dowling et al. [17]. Specifically, they introduced a distinct intermediate IND-1CCA-MAC game to demonstrate that OW-CPA KEMs are sufficient for TLS 1.3 in the ROM. They first proved that OW-CPA KEMs imply the security of the IND-1CCA-MAC with a secure MAC in the ROM, then utilized the security of the IND-1CCA-MAC to prove the security of TLS 1.3 in the standard model. Notably, the IND-1CCA-MAC game only serves as an intermediate step in the proof.

However, they only proved that OW-CPA KEMs can derive IND-1CCA-MAC security with a secure MAC in the ROM [22]. They did not extend their proof to the QROM. Also, the bound of the ROM proof is very much non-tight, with $\epsilon_R \approx O(1/q^6)\epsilon_{\mathcal{A}}$ for OW-CPA KEMs to prove IND-1CCA-MAC secure, where ϵ_R (resp. $\epsilon_{\mathcal{A}}$) is the advantage of the reduction R (resp. adversary $\mathcal{A}$) breaking the OW-CPA security of the underlying KEM (resp. the IND-1CCA-MAC security), and q is the number of $\mathcal{A}$'s queries to the random oracle (RO). Therefore, they suggest that due to the weak security bound associated with OW-CPA KEMs, employing IND-1CCA KEMs might be more advantageous in the PQ TLS 1.3 handshake because the security bound for IND-1CCA KEMs provides better guarantees than OW-CPA KEMs. Consequently, better parameters can be chosen for the implementation of the underlying IND-1CCA KEMs. Furthermore, they leave the development of tighter ROM reductions and proofs in the QROM as open problems, as stated in Section 4.3 of their paper [22].

The overall bound for TLS security from OW-CPA is very much non-tight. This is clearly not sufficient to guarantee security in practice, and we leave the improvement of the bounds as an interesting open question and leave security in the QROM as future work.

1.1 Our Contributions

We resolve the open problem by introducing a new intermediate security game IND-1CCA-MAC* (a variant of the IND-1CCA-MAC [22] and suitable for establishing the security of TLS 1.3 handshake), and by reducing the CPA security of the underlying KEM to IND-1CCA-MAC* in a tighter manner. In particular, our results show that standard CPA-secure KEMs are sufficient to guarantee the security of TLS 1.3 in practice. Our main contributions are as follows:

1. First, we prove the security of IND-1CCA-MAC* from standard CPA-secure KEMs in the ROM. Specifically, our reduction exhibits a tightness of $\epsilon_R \approx O(1/q)\epsilon_{\mathcal{A}}$, which is much tighter than $\epsilon_R \approx O(1/q^6)\epsilon_{\mathcal{A}}$ given by [22] (see Table 1). Such a CPA-secure KEM can be directly obtained by instantiating the PKE scheme in Fig. 1 with the one used in CRYSTALS-Kyber [10].

2. Moreover, we also show that for rigid D-OW-CPA KEMs the reduction can be tight, with $2\epsilon_R \approx \epsilon_{\mathcal{A}}$. Here, rigid D-OW-CPA KEMs denote KEMs that are constructed by applying a simple transform to a rigid one-way secure deterministic PKE [4], as shown in Fig. 1. In particular, the NIST-PQC Round-3 Finalist NTRU [13] and the NIST-PQC Round-4 Candidate Classic McEliece [2] are based on rigid one-way secure deterministic PKEs, which can be transformed into the corresponding D-OW-CPA KEMs in this paper.
3. Then, we first prove the security of IND-1CCA-MAC* from OW-CPA/IND-CPA/D-OW-CPA [5] KEMs in the QROM. In particular, Huguenin-Dumittan and Vaudenay [22] conjectured that the compressed oracle technique introduced by Zhandry [39] could be useful in the QROM proof due to their extensive use of the programming property of ROs in the ROM proof. However, in our QROM proof, we only utilize two other well-established techniques: one-way to hiding (OW2H) [3, 8] and measure-and-reprogram [14, 15, 24]. Specifically, our reduction achieves a tightness of $\epsilon_R \approx O(1/q^2)\epsilon_A^2$ in the QROM with IND-CPA/D-OW-CPA secure KEMs.
4. Finally, we show that if the IND-1CCA-MAC* security is satisfied, then the MultiStage [17] security of TLS 1.3 handshake protocol is satisfied in the standard model. In particular, the reduction for TLS 1.3 from IND-1CCA-MAC* exhibits the same tightness as the reduction given by [22] for TLS 1.3 from IND-1CCA-MAC or 1CCA KEM. Putting everything together, we finally prove that if the underlying KEM is OW-CPA/IND-CPA/D-OW-CPA secure, then the TLS 1.3 handshake protocol is secure in the MultiStage model with a much tighter ROM proof and the first QROM proof.

Remark 1. Our results show that the reduction bounds for IND-1CCA-MAC* from CPA KEMs exhibit the same tightness as those 1CCA KEMs from PKEs [24]. We also note that CPA KEMs in Fig. 1 can be tightly reduced to CPA PKEs. Therefore, if we consider the complete reduction from the CPA security of the underlying PKE to the MultiStage security of the resulting TLS 1.3, our reduction for TLS 1.3 with CPA KEM has the same tightness as the currently tightest reduction for TLS 1.3 with 1CCA KEM given by [24]. Notably, the CPA-secure KEM used in this paper is more concise and efficient compared to the currently employed CCA/1CCA-secure KEM.

[4] The rigid [7] property means that decrypting a ciphertext c and then re-encrypting yields c. For a general deterministic PKE, the rigid property can be achieved through a re-encryption transform.

[5] In the QROM, we do not require the rigid property for D-OW-CPA KEMs because the simulations of $\mathsf{O}^{\mathrm{Dec}}$ and $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ are the same as standard CPA KEMs in the proof.

[6] All proofs for IND-1CCA-MAC* from CPA KEMs rely on a secure MAC, we focus on the primary KEM component here. In [22], they actually prove IND-1CCA-MAC, which implies the security of IND-1CCA-MAC*. Essentially, IND-1CCA-MAC* is sufficient for the security proof of TLS 1.3, see Theorem 6 for details.

Table 1: Reduction tightness of the intermediate game IND-1CCA-MAC*.

Underlying KEM	Reduction Tightness[6]	Model
OW-CPA [22]	$\epsilon_R \approx O(1/q^6)\epsilon_A$	ROM
OW-CPA (Our work)	$\epsilon_R \approx O(1/q^2)\epsilon_A$	ROM
IND-CPA (Our work)	$\epsilon_R \approx O(1/q)\epsilon_A$	ROM
D-OW-CPA (Our work)	$\epsilon_R \approx O(1)\epsilon_A$	ROM
OW-CPA [22]	-	QROM
OW-CPA (Our work)	$\epsilon_R \approx O(1/q^4)\epsilon_A^2$	QROM
IND-CPA (Our work)	$\epsilon_R \approx O(1/q^2)\epsilon_A^2$	QROM
D-OW-CPA (Our work)	$\epsilon_R \approx O(1/q^2)\epsilon_A^2$	QROM

1.2 Practical Efficiency Impact

As shown in Fig. 2, the most economical method to construct a secure TLS 1.3 is to use an OW-CPA/IND-CPA secure KEM, obviating the need for transformations like FO or T_{CH}, T_H, T_{RH} to achieve 1CCA security. Directly using the CPA KEM based on CRYSTALS-Kyber.PKE [31] as in Fig. 1 can bring a significant speed improvement, see Table 2. In particular, for decapsulation, there is a 6X speedup over using T_{CH}, T_{RH}, and a 20X speedup over using FO.

Encaps(pk)	Decaps(sk, c)
1 : $m \leftarrow\!\$\ \mathcal{M}$	1 : $m' = \mathrm{dec}'(\mathsf{sk}, c)$
2 : $c \leftarrow\!\$\ \mathrm{enc}'(\mathsf{pk}, m)$	2 : **if** $m' = \perp$
3 : $K := m$ //CPA	3 : **return** $\perp$
4 : $K := H(m, c)$ //1CCA	4 : **else return** $K := m'$ //CPA
5 : **return** (K, c)	5 : **else return** $K := H(m', c)$ //1CCA

Fig. 1: The construction of CPA and 1CCA (T_H [22]) KEMs from PKEs

Table 2: Benchmark of Encaps and Decaps for CRYSTALS-Kyber [31] with different transforms using liboqs (AVX2 enabled, NIST security level I) on system specs: Intel(R) Core(TM) i9-10900X CPU @ 3.7 GHz, 32.0 GB RAM, 64-bit OS.

Algorithm	CPA	T_{RH}	T_{CH}	FO
Encaps (μs)	5.35	7.255	7.682	7.666
Decaps (μs)	0.366	2.274	2.277	7.428

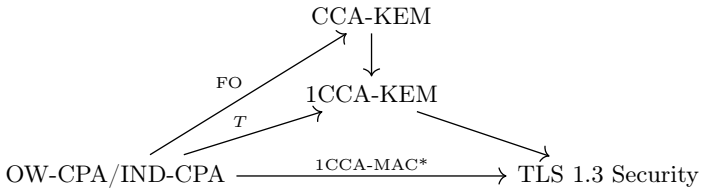

Fig. 2: Diagram of the process for constructing TLS 1.3 using KEMs based on different security assumptions. This figure shows that the most straightforward method to construct a secure TLS 1.3 is to use an OW-CPA/IND-CPA secure KEM, which does not require the redundant T transformation [22, 24] or FO transformation [18, 20]. Here, T represents a general term for T_{CH}, T_H, T_{RH}.

1.3 Technique Overview

Approach by Huguenin-Dumittan and Vaudenay [22]: To demonstrate that OW-CPA KEMs are sufficient for the security of TLS1.3 (the description of the TLS 1.3 protocol is illustrated in Fig. 5.), Huguenin-Dumittan and Vaudenay introduced a special intermediate IND-1CCA-MAC game, as depicted in Fig. 4. They first demonstrated that an OW-CPA KEM with a secure MAC implies IND-1CCA-MAC security and then established the security of TLS 1.3 based on the IND-1CCA-MAC security. Specifically, the reduction bounds for deriving TLS 1.3 security from IND-1CCA-MAC KEMs or from IND-1CCA KEMs are both $O(t_s^2\epsilon_{\mathcal{A}} + t_s t_u \epsilon_{\mathcal{B}}^{\mathrm{SIG}})$, where t_s (resp. t_u) is the maximal number of sessions (resp. users) and $\epsilon_{\mathcal{A}}$ (resp. $\epsilon_{\mathcal{B}}^{\mathrm{SIG}}$) is the advantage of $\mathcal{A}$ (resp. $\mathcal{B}$) breaking the IND-1CCA/IND-1CCA-MAC security for KEM (resp. EUF-CMA security for signature). The following is an overview of the rationale behind using this special IND-1CCA-MAC game to prove the security of TLS 1.3. When attempting to prove the security of TLS 1.3, it is necessary to simulate the specific entire session and the queries by the adversary. Specifically, when the tested session is the server session labeled label_S, and its partner is labeled label_C. If the Server Hello (SH) message (including the original SH and Server Key Share (SKS)) sent by label_S differs from the SH message received by label_C, it indicates potential tampering by the adversary with SH values. However, the reduction must still simulate the honest partner label_C to complete the handshake protocol and answer the adversary's queries. Therefore, by utilizing two oracles, $\mathsf{O}^{\mathrm{Dec}}$ and $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ in the IND-1CCA-MAC game, the reduction can perfectly simulate this scenario. Specifically, the reduction can query $\mathsf{O}^{\mathrm{Dec}}$ to obtain stage-1 and stage-2 keys tk_C, tk_S, enabling perfect simulation of label_C and any Reveal queries up to the Server Finished (SF) message. Upon label_C receiving the SF message, which contains a MAC tag, the reduction can query $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ to verify this tag. If this tag is correct, the reduction obtains the Handshake Secret (HS) and can derive all necessary secrets to perfectly simulate label_C.

Causes of Reduction Loss in [22]: When proving that OW-CPA KEMs are IND-1CCA-MAC secure in the ROM, the reduction needs to simulate

$\mathsf{O}^{\mathrm{Dec}}(\overline{\mathrm{ct}}, \overline{n})$, $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}(\mathrm{ct}, n, \mathrm{tag}, \mathrm{txt})$ without secret key and embed the underlying security experiment into the IND-1CCA-MAC instance. When simulating $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$, the reduction randomly takes one of adversary's query $H_2(\mathrm{HS}, t)$ (corresponding to $H_2(\mathrm{HS}, H_T(\mathrm{ct}, n))$ in $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$) or $\perp$ as the return, with the success probability of guessing correctly being $O(1/(q_{H_2}+1))$. When simulating $\mathsf{O}^{\mathrm{Dec}}$, the reduction takes one of the adversary's query $H_1(\mathrm{HS}, t), H_2(\mathrm{HS}, t)$ (corresponds to $H_1(\overline{\mathrm{HS}}, H_T(\overline{\mathrm{ct}}, \overline{n}))$, $H_2(\overline{\mathrm{HS}}, H_T(\overline{\mathrm{ct}}, \overline{n}))$ in $\mathsf{O}^{\mathrm{Dec}}$) or $\perp$ as the return, or $\perp_d$ (this guess indicates that the adversary has not queried about the corresponding value $H_1(\overline{\mathrm{HS}}, H_T(\overline{\mathrm{ct}}, \overline{n}))$, $H_2(\overline{\mathrm{HS}}, H_T(\overline{\mathrm{ct}}, \overline{n}))$ and $\mathrm{decaps}(\mathsf{sk}, \overline{\mathrm{ct}}) \neq \perp$), with the success probability of guessing correctly being $O(1/(q_{H_1}+2)(q_{H_2}+2))$. In the case of $\perp_d$, the reduction randomly chooses $\overline{chts}, \overline{shts}$, and finally returns $H_D(\overline{chts}), H_D(\overline{shts})$ in $\mathsf{O}^{\mathrm{Dec}}$. After this, the reduction must ensure the consistency of H_1 and H_2 with $\mathsf{O}^{\mathrm{Dec}}$ by guessing whether $H_1(\mathrm{HS}, t), H_2(\mathrm{HS}, t)$ corresponds to the potentially defined $\overline{chts}, \overline{shts}$ in $\mathsf{O}^{\mathrm{Dec}}$, with the success probability of guessing correctly being $O(1/(q_{H_1}+1)(q_{H_2}+1))$. When embedding the instance of the underlying OW-CPA experiment into the IND-1CCA-MAC instance, an OW-CPA instance is embedded with a $O(1/q_G)$ loss in the ROM. Thus, the total loss is $O(1/q^6)$ in the ROM.

Below, we elaborate on how to improve the reduction of the above loss in the ROM and lift our tighter ROM proof into the QROM setting.

A New Intermediate Game: IND-1CCA-MAC*: We observe that we only need a specific IND-1CCA-MAC game denoted as IND-1CCA-MAC* to prove the security of TLS1.3. IND-1CCA-MAC* is identical to IND-1CCA-MAC, except it constrains the adversary $\mathcal{A}$ to initiating the first query to $\mathsf{O}^{\mathrm{Dec}}(\overline{\mathrm{ct}}, \overline{n})$ and subsequent query to $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}(\mathrm{ct}, n, \mathrm{tag}, \mathrm{txt})$ with $(\mathrm{ct}, n) = (\overline{\mathrm{ct}}, \overline{n})$. This restriction in IND-1CCA-MAC* does not impact the proof for TLS1.3 because in the proof when the adversary send a forged $\mathrm{SH} = (\mathrm{ct}, n_s)$ message the reduction just queries $\mathsf{O}^{\mathrm{Dec}}(\mathrm{ct}, n_S)$ first and $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}(\mathrm{ct}, n_S, \mathrm{txt}, \mathrm{tag})$ later to perfectly simulate the game. Specifically, following the method in [22], we can easily utilize IND-1CCA-MAC* to prove the MultiStage security of TLS 1.3. In particular, by utilizing IND-1CCA-MAC*, we can employ the guess from $\mathsf{O}^{\mathrm{Dec}}$ to perfectly simulate $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$, thereby avoiding the $O(1/(q_{H_2}+1))$ security loss in the ROM and this new intermediate game also plays a crucial role for the QROM proof.

Combining RO Simulation Technique: When there are two ROs, $H_1(x)$ and $H_2(x)$, with identical input space, we need to operate on these two ROs on the same input x_0, which could be either reprogramming or guessing which query to $H_1(\cdot), H_2(\cdot)$ corresponds to x_0. In this paper, we combine these two ROs by defining $H_{12} = (H_1, H_2)$ to simulate H_1 and H_2 simultaneously. In this scenario, the total number of queries to H_{12} is at most $q_{H_1} + q_{H_2}$. Thus operating only on H_{12} may result in a tighter reduction loss. More precisely, they have to guess separately for the two ROs, resulting in a loss of $O(1/(q_{H_1}+1)(q_{H_2}+1))$ in [22]. By using the combining RO simulation technique, we only need to make one guess for $H_{12} = (H_1, H_2)$, thereby reducing the security loss to $O(1/(q_{H_1}+q_{H_2}+1))$.

Specifically, to simulate the $\mathsf{O}^{\mathrm{Dec}}(\overline{\mathrm{ct}}, \overline{n})$ oracle without the secret key, we initially employ an internal hash function $H_{12} = (H_1, H_2)$ to simulate the ran-

dom oracles H_1 and H_2. We first randomly choose $guess \leftarrow\$ \{0,1\}$ to guess whether $\mathsf{decaps}(\overline{\mathrm{ct}}, \mathsf{sk}) = \perp$. If $guess = 0$, we return $\perp$ in $\mathsf{O}^{\mathrm{Dec}}$. Otherwise, for a valid ciphertext $\overline{\mathrm{ct}}$ such that $\perp \neq K \leftarrow \mathsf{decaps}(\mathsf{sk}, \overline{\mathrm{ct}})$, the Dec oracle should return $(H_D(H_1(\overline{\mathrm{HS}}, \bar{t})), H_D(H_2(\overline{\mathrm{HS}}, \bar{t})))$, where $\overline{\mathrm{HS}} = G(K), \bar{t} = H_T(\overline{\mathrm{ct}}, \overline{n})$. Consequently, following [24], we directly reprogram $H_{12}(\overline{\mathrm{HS}}, \bar{t})$ with random values $\Theta = (\overline{chts}, \overline{shts})$, and then output $H_D(\overline{chts}), H_D(\overline{shts})$ in the $\mathsf{O}^{\mathrm{Dec}}$ oracle. Intuitively, the simulation is perfect if we reprogram $H_{12}(\overline{\mathrm{HS}}, \bar{t})$ with Θ when the adversary $\mathcal{A}$ first queries $(\overline{\mathrm{HS}}, \bar{t})$ to either H_1 or H_2. In the ROM, a random guess is correct with a probability of $1/(q_{H_1} + q_{H_2} + 1)$.

Utilizing IND-CPA: When embedding the underlying security experiment into the IND-1CCA-MAC* instance, we successfully embed an IND-CPA instance in the ROM without reduction loss, and an OW-CPA instance is embedded in the ROM with an $O(1/q)$ loss.

Rigid D-OW-CPA: We can utilize rigid D-OW-CPA KEMs to prove IND-1CCA-MAC*/IND-1CCA-MAC tightly. The fundamental reason is that with the rigid property of the underlying deterministic PKE, we can perfectly simulate $\mathrm{HS} = G(\mathsf{decaps}(\mathsf{sk}, \mathrm{ct}))$ without sk when $\mathsf{decaps}(\mathsf{sk}, \mathrm{ct}) \neq \perp$ if G is a random oracle. Thus we can use HS perfectly simulate $\mathsf{O}^{\mathrm{Dec}}$, $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$. Moreover, we embed a D-OW-CPA instance in the ROM without reduction loss.

Specifically, we note that the δ-correctness implies that all queries to $G(K)$ are keys K that do not induce correctness errors. When the adversary queries $G(K)$, we return HS by lazy sampling and updating $\mathfrak{L} = \mathfrak{L} \cup (K, \mathrm{ct}, \mathrm{HS})$, where $\mathrm{ct} = \mathsf{enc}'(K)$. Subsequently, when the adversary queries $\mathsf{O}^{\mathrm{Dec}}(\mathrm{ct}_1, n_1)$ (resp. $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}(\mathrm{ct}_2, n_2, \mathrm{tag}, \mathrm{txt})$), we first randomly choose $guess \leftarrow\$ \{0,1\}$ to guess whether $\mathsf{decaps}(\mathrm{ct}_1, \mathsf{sk}) = \perp$ (resp. $\mathsf{decaps}(\mathrm{ct}_2, \mathsf{sk}) = \perp$). When $guess = 0$, we return $\perp$. When $guess = 1$ and the corresponding ct_1 (resp. ct_2) exists in $\mathfrak{L}$, we directly extract the HS corresponding to ct_1 (resp. ct_2) from $\mathfrak{L}$. Otherwise, we can conclude that the adversary has not yet queried $G(K)$, where $K = \mathsf{dec}'(\mathsf{sk}, \mathrm{ct}_1)$ (resp. $K = \mathsf{dec}'(\mathsf{sk}, \mathrm{ct}_2)$). Assuming the adversary has queried $G(K)$ before, this implies the existence of $(K, \mathrm{ct}'_1, \cdot) \in \mathfrak{L}$, where $\mathsf{enc}'(\mathsf{pk}, K) = \mathrm{ct}'_1$. According to the rigid property, we have $\mathrm{ct}_1 = \mathrm{ct}'_1$, which contradicts the condition. Therefore, we just directly sample a random HS and record $\mathfrak{L}_{\mathrm{Dec}} = \{\mathrm{ct}_1, \mathrm{HS}\}$ (resp. $\mathfrak{L}^{\mathrm{MAC}}_{\mathrm{Dec}} = \{\mathrm{ct}_2, \mathrm{HS}\}$). Finally, using HS we can directly simulate $\mathsf{O}^{\mathrm{Dec}}$ (resp. $\mathsf{O}^{\mathrm{MAC}}_{\mathrm{Dec}}$). To ensure consistency between the random oracle G and $\mathsf{O}^{\mathrm{Dec}}$ (resp. $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$), in the simulation of following $G(K)$, we first compute $\mathrm{ct} = \mathsf{enc}'(K)$ and directly return HS from $\mathfrak{L}_{\mathrm{Dec}}$ (resp. $\mathfrak{L}^{\mathrm{MAC}}_{\mathrm{Dec}}$) if $\mathrm{ct} = \mathrm{ct}_1$ (resp. $\mathrm{ct} = \mathrm{ct}_2$). If $\mathrm{ct} \neq \mathrm{ct}_1$ (resp. $\mathrm{ct} \neq \mathrm{ct}_2$), we can assert that $\mathsf{dec}'(\mathsf{sk}, \mathrm{ct}) \neq \mathsf{dec}'(\mathsf{sk}, \mathrm{ct}_1)$ (resp. $\mathsf{dec}'(\mathsf{sk}, \mathrm{ct}) \neq \mathsf{dec}'(\mathsf{sk}, \mathrm{ct}_2)$) based on the rigid property. Therefore, we can perfectly simulate $\mathsf{O}^{\mathrm{Dec}}$ and $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ when we guess correctly, and the probability of guessing correctly in each case is $1/2$. Lastly, the D-OW-CPA adversary can directly identify a K such that $\mathsf{enc}'(K) = \mathrm{ct}^*$ in the G-List and return K as the solution to the D-OW-CPA instance without random guessing, thereby embedding a D-OW-CPA instance without reduction loss.

QROM: We have discussed the primary techniques used to achieve a tighter security reduction for IND-1CCA-MAC* in the ROM. However, achieving QROM proof for IND-1CCA-MAC* from OW-CPA/IND-CPA/D-OW-CPA is still challenging. The fundamental difficulty in the proof revolves around embedding the underlying security experiment into the IND-1CCA-MAC* instance and simulate $\mathsf{O}^{\mathrm{Dec}}$, $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ without the secret key. We delve into how we solve each of these challenges in detail.

To embed the instance of the underlying security experiment into the IND-1CCA-MAC* instance, if the underlying KEM is OW-CPA secure, we follow previous proofs [8, 24, 25] and use general OW2H to argue the embedding of the underlying instance. Moreover, if the KEM is D-OW-CPA, we employ double-sided OW2H to discuss the implications of instance embedding. Similarly, if the KEM is IND-CPA, we utilize double-sided OW2H and extended double-sided OW2H [24] to discuss the implications of instance embedding.

To simulate the $\mathsf{O}^{\mathrm{Dec}}(\overline{\mathrm{ct}}, \overline{n})$ oracle without the secret key, we initially utilize an internal hash function $H_{12} = (H_1, H_2)$ to simulate the random oracles H_1 and H_2. Firstly, we randomly choose $guess \leftarrow\!\$\ \{0,1\}$ to guess whether $\mathrm{decaps}(\mathrm{ct}, \mathsf{sk}) = \perp$. If $guess = 0$, we return $\perp$ in $\mathsf{O}^{\mathrm{Dec}}$. Otherwise, for a valid ciphertext $\bar{c}$ such that $\perp \neq K \leftarrow \mathrm{decaps}(\mathsf{sk}, \bar{c})$, the Dec oracle should return $H_D(H_1(\overline{\mathrm{HS}}, \bar{t})), H_D(H_2(\overline{\mathrm{HS}}, \bar{t}))$, where $\overline{\mathrm{HS}} = G(K)$ and $\bar{t} = H_T(\overline{\mathrm{ct}}, \overline{n})$. As discussed in the ROM, we directly reprogram $H_{12}(\overline{\mathrm{HS}}, \bar{t})$ to random values $\Theta = (\overline{chts}, \overline{shts})$, and then output $H_D(\overline{chts}), H_D(\overline{shts})$ in the $\mathsf{O}^{\mathrm{Dec}}$ oracle. Intuitively, the simulation is perfect if we reprogram $H_{12}(\overline{\mathrm{HS}}, \bar{t})$ to Θ when the adversary $\mathcal{A}$ first queries $(\overline{\mathrm{HS}}, H_T(\overline{\mathrm{ct}}, \overline{n}))$ to either H_1 or H_2. In the ROM, the probability of a correct random guess is $1/q_{H_{12}}$. However, in the QROM, since the adversary's queries are superposition RO-queries, it is difficult to define when the adversary makes a query $H_{12}(\overline{\mathrm{HS}}, \bar{t})$ to either H_1 or H_2. Therefore, in the QROM, we adopt the approach from [24] to argue it differently. We observe that the consistency between $\mathsf{O}^{\mathrm{Dec}}$ and H_{12} can be ensured if the predicate $\mathsf{O}^{\mathrm{Dec}}(\overline{\mathrm{ct}}, \overline{n}) = (H_D(H_1(\overline{\mathrm{HS}}, \bar{t})), H_D(H_2(\overline{\mathrm{HS}}, \bar{t})))$ holds true. In the practical implementation of the decryption oracle $\mathsf{O}^{\mathrm{Dec}}(\overline{\mathrm{ct}}, \overline{n})$, there is an implicit classical query to H_{12}, which is omitted during the oracle's simulation in $\mathsf{O}^{\mathrm{Dec}}$. Therefore, we utilize the refined optional query measure-and-reprogram technique [24], as outlined in Lemma 6, to argue this simulation impact.

Now we consider how to simulate the $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}(\overline{\mathrm{ct}}, \overline{n}, \mathrm{txt}, \mathrm{tag})$ oracle without the secret key. Note that $(\overline{\mathrm{ct}}, \overline{n}, \cdot, \cdot)$ remains consistent with the $\mathsf{O}^{\mathrm{Dec}}(\overline{\mathrm{ct}}, \overline{n})$ oracle, as defined in the IND-1CCA-MAC* game. If the $guess$ value in $\mathsf{O}^{\mathrm{Dec}}$ is 0, we correspondingly return $\perp$ in $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$. Otherwise, for a valid ciphertext $\bar{c}$, in the ROM, if $\mathrm{MAC.Vrf}(fk_S, \mathrm{txt}, \mathrm{tag})$ is correct, where $fk_S = H_D(\overline{shts})$, we can assert that the adversary has already queried the corresponding H_2 with high probability. Subsequently, using the guess i from the simulation of $\mathsf{O}^{\mathrm{Dec}}$, we extract the $i+1$-th query directly and output the respective HS_{i+1}.

In the QROM, we simulate $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ by combining the refined optional-query measure-and-reprogram technique and OW2H technique. During the simulation of $\mathsf{O}^{\mathrm{Dec}}(\overline{\mathrm{ct}}, \overline{n})$, we have utilized the refined optional-query measure-and-

reprogram technique to H_{12}. Specifically, we measure the input quantum state to obtain x and subsequently reprogramming H_{12} at x to Θ conditioned on some random values. If MAC.$\underline{\mathrm{Vrf}}(fk_S$, txt, tag) is correct, we need to return HS in $\mathsf{O}^{\mathrm{Dec}}$, where $fk_S = H_D(\overline{shts})$. Intuitively, we can use x (which includes HS) to directly output HS. However, it is important to discuss the order of $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ query and the measurement in the optional-query measure-and-reprogram technique. If $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ query occurs after the measurement, we can easily use x to perfectly simulate the output of HS. However, if $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ query occurs before the measurement, we cannot return HS since we have not yet measured it to obtain x. Nonetheless, this scenario implies that all H_{12} queries made by the adversary have not been reprogrammed and, consequently, are unrelated to the reprogrammed values Θ, which is utilized to derive the MAC key. Therefore, we can argue that the adversary $\mathcal{A}$ cannot distinguish the MAC key fk_S from a random value using the OW2H Lemma. Consequently, the adversary cannot forge a valid tag based on the security of MAC. Therefore, if $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ query occurs before the measurement, we directly output $\perp$ in $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$.

2 Preliminaries

2.1 Notation

The security parameter is denoted as λ. The set $\{0, \ldots, q\}$ is denoted as $[q]$. PPT is denoted to represent probabilistic polynomial time. $\mathcal{K}$, $\mathcal{M}$ and $\mathcal{C}$ denote the key space, message space, and ciphertext space, respectively. For a finite set X, $x \leftarrow\!\!\$\ X$ represents the sampling of a uniformly random element from X. $\Pr[P : G]$ indicates the probability that the predicate holds true when free variables in P are assigned according to the program in G. The sampling from some distribution D is represented by $x \leftarrow\!\!\$\ D$. For a quantum or randomized classical algorithm (resp. deterministic) A, $y \leftarrow\!\!\$\ A(x)$ (resp. $y \leftarrow A(x)$) denotes that A outputs y on input x. $x =?y$ is denoted as an integer that is 1 if $x = y$, and 0 otherwise. $|X|$ denotes the cardinality of set X. A^H (resp. $A^{|H\rangle}$) denotes that algorithm A gains classical (resp. quantum) access to the oracle H.

2.2 Cryptographic Primitives

Definition 1 (Deterministic Public-Key Encryption). *A DPKE over $\mathcal{M}$ is a tuple of three algorithms* gen, enc, dec. *(1)* $(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathrm{gen}(1^\lambda)$*: The key generation algorithm* gen *takes as inputs the security parameter and outputs a key pair* $(\mathsf{pk}, \mathsf{sk})$*. Usually, we will omit the input of* gen *for brevity. (2)* $\mathrm{ct} \leftarrow \mathrm{enc}(\mathsf{pk}, m)$*: The encryption algorithm takes as inputs the public key* pk *and a message $m \in \mathcal{M}$ and deterministically outputs a ciphertext* ct*. (3)* $m' \leftarrow \mathrm{dec}(\mathsf{sk}, \mathrm{ct})$*: The decryption algorithm, on input the secret key* sk *and the ciphertext* ct*, deterministically outputs a message $m' \in \mathcal{M} \cup \{\perp\}$.*

Correctness. A Deterministic Public-Key Encryption (DPKE) is δ-correct if $E\left[\max_{m\in M}\Pr[\mathrm{dec}(\mathsf{sk},\mathrm{ct}) \neq m : \mathrm{ct} \leftarrow \mathrm{enc}(\mathsf{pk}, m)]\right] \leq \delta$, where the expectation is taken over $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathrm{gen}$. We say a DPKE is perfectly correct if $\delta = 0$.
Rigidity. [7] [7] A DPKE is *rigid* if for all key pairs $(\mathsf{pk},\mathsf{sk}) \leftarrow\!\!\$\ \mathrm{gen}$, and all ciphertexts ct, it holds that either $\mathrm{dec}(\mathsf{sk},\mathrm{ct}) = \perp$ or $\mathrm{enc}(\mathsf{pk},\mathrm{dec}(\mathsf{sk},\mathrm{ct})) = \mathrm{ct}$.

Definition 2 (OW-CPA-secure DPKE). *A DPKE scheme* $\mathrm{DPKE} = (\mathrm{gen},\mathrm{enc},\mathrm{dec})$ *is* OW-CPA *if for any PPT adversary* $\mathcal{A}$ *we have*

$$\begin{aligned}\mathrm{Adv}^{\text{OW-CPA}}_{\mathrm{DPKE}}(\mathcal{A}) &= \Pr\left[\mathcal{A}(\mathsf{pk},\mathrm{ct}^*) \Rightarrow m^* : \begin{array}{c}(\mathsf{pk},\mathsf{sk}) \leftarrow\!\!\$\ \mathrm{gen};\\ m^* \leftarrow\!\!\$\ \mathcal{M}; \mathrm{ct}^* = \mathrm{enc}(\mathsf{pk}, m^*)\end{array}\right]\\ &= \mathit{negl}(\lambda),\end{aligned}$$

where the probability is taken over the randomness of the public-key generation and the adversary $\mathcal{A}$*.*

Definition 3 (Key Encapsulation Mechanism). *A KEM over* $\mathcal{K}$ *is a tuple of three algorithms* gen, encaps, decaps*. (1)* $(\mathsf{pk},\mathsf{sk}) \leftarrow\!\!\$\ \mathrm{gen}(1^\lambda)$*: The key generation algorithm* gen *takes as inputs the security parameter and outputs a key pair* $(\mathsf{pk},\mathsf{sk})$*. Usually, we will omit the input of* gen *for brevity. (2)* $(\mathrm{ct}, K) \leftarrow\!\!\$\ \mathrm{encaps}(\mathsf{pk})$*: The encapsulation algorithm takes as inputs the public key* pk *and it outputs a tuple* (ct, K)*, where* $K \in \mathcal{K}$ *and* $\mathrm{ct} \in \mathcal{C}$*. (3)* $K' \leftarrow \mathrm{decaps}(\mathsf{sk},\mathrm{ct})$*: The decapsulation procedure, on input the secret key* sk *and the ciphertext* ct*, outputs a key* K'*. If the KEM allows explicit rejection, the output is a key* $K' \in \mathcal{K}$ *or the rejection symbol* $\perp$*.*

Definition 4 (OW-CPA-secure KEM). *A KEM scheme* $\mathrm{KEM} = (\mathrm{gen},\mathrm{encaps},\mathrm{decaps})$ *is* OW-CPA *if for any PPT adversary* $\mathcal{A}$ *we have*

$$\begin{aligned}\mathrm{Adv}^{\text{OW-CPA}}_{\mathrm{KEM}}(\mathcal{A}) &= \Pr\left[\mathcal{A}(\mathsf{pk},\mathrm{ct}^*) \Rightarrow K^* : \begin{array}{c}(\mathsf{pk},\mathsf{sk}) \leftarrow\!\!\$\ \mathrm{gen};\\ (K^*,\mathrm{ct}^*) \leftarrow\!\!\$\ \mathrm{encaps}(\mathsf{pk})]\end{array}\right]\\ &= \mathit{negl}(\lambda),\end{aligned}$$

where the probability is taken over the randomness of the public-key generation, encapsulation, and the adversary $\mathcal{A}$*.*

It is straightforward to construct an OW-CPA-secure KEM = (gen, encaps, decaps) based on an OW-CPA-secure DPKE = (gen′, enc′, dec′) using the simple CPA transform shown in Fig. 1. In this paper, we denote this particular construction of KEM as DKEM or D-OW-CPA KEM.

[7] The NIST-PQC Round-3 Finalist NTRU [13] and NIST-PQC Round-4 Candidate Classic McEliece [2], are based on rigid one-way secure deterministic PKEs. For a general deterministic PKE, the rigid property can be achieved through a re-encryption transform.

Definition 5 (IND-CPA-secure KEM). *We define the* IND-CPA *game for KEM as in Fig. 3. A KEM scheme* $\mathrm{KEM} = (\mathrm{gen}, \mathrm{encaps}, \mathrm{decaps})$ *is* IND-CPA *if for any PPT adversary* $\mathcal{A}$ *we have* $\mathrm{Adv}^{\text{IND-CPA}}_{\mathrm{KEM}}(\mathcal{A}) := \left|\Pr[\text{IND-CPA}_{\mathrm{KEM}}(\mathcal{A}) \Rightarrow 1] - \frac{1}{2}\right| = negl(\lambda)$.

Game $\text{IND-CPA}_{\mathrm{KEM}}(\mathcal{A})$	
1 :	$(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathrm{gen}$
2 :	$b \leftarrow\!\!\$\ \{0,1\}$
3 :	$(K_0^*, \mathrm{ct}^*) \leftarrow\!\!\$\ \mathrm{encaps}(\mathsf{pk})$
4 :	$K_1^* \leftarrow\!\!\$\ \mathcal{K}$
5 :	$b' \leftarrow \mathcal{A}(\mathsf{pk}, \mathrm{ct}^*, K_b^*)$
6 :	**return** $b' =? b$

Fig. 3: IND-CPA game for KEM.

Definition 6 (MAC EUF-0T). *Let* $\mathrm{MAC} = (\mathrm{MAC.Vrf}, \mathrm{MAC.Tag})$ *be a message authentication code scheme (MAC). We say* MAC *is* EUF-0T *if for any PPT adversary* $\mathcal{A}$, $\mathrm{Adv}^{\mathrm{MAC}}_{\text{EUF-0T}}(\mathcal{A}) := \Pr[\mathrm{MAC.Vrf}(K, m, tag) = 1 : (m, tag) \leftarrow \mathcal{A}; K \leftarrow \mathcal{K}]$ *is negligible in the security parameter, where the probability is taken over the sampling of the key and the randomness of the adversary.*

Definition 7 (IND-1CCA-MAC [22]). *We consider the games defined in Fig. 4. Let* $\mathcal{K}$ *be the key space,* G, H_1, H_2, H_3, H_4, *and* H_D *be key-derivation functions with images in* $\{0,1\}^n$, H_T *be a hash function with images in* $\{0,1\}^n$, *and a MAC scheme* MAC. *A KEM scheme* $\mathrm{KEM} = (\mathrm{gen}, \mathrm{encaps}, \mathrm{decaps})$ *is* IND-1CCA-MAC *if for any PPT adversary* $\mathcal{A}$ *we have*

$$\mathrm{Adv}^{\text{IND-1CCA-MAC}}_{\mathrm{KEM}}(\mathcal{A}) := \left|\Pr[\text{IND-1CCA-MAC}_{\mathrm{KEM}}(\mathcal{A}) \Rightarrow 1] - \frac{1}{2}\right| = \mathrm{negl}(\lambda)$$

where $\Pr[\text{IND-1CCA-MAC}^b_{\mathrm{KEM}}(\mathcal{A}) \Rightarrow 1]$ *is the probability that* $\mathcal{A}$ *wins the* $\text{IND-1CCA-MAC}^b_{\mathrm{KEM}}(\mathcal{A})$ *game defined in Fig. 4.*

In this game, the adversary receives a public key, a challenge ciphertext ct^* encapsulating a key K^*, a nonce n^*, and access two oracles, $\mathsf{O}^{\mathrm{Dec}}$ and $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$, which it can query at most once to distinguish between three secrets: $(\mathrm{CHTS}_0, \mathrm{SHTS}_0, \mathrm{dHS}_0)$ derived from K^*, ct^* and n^*, or three random secrets $(\mathrm{CHTS}_1, \mathrm{SHTS}_1, \mathrm{dHS}_1)$. It is important to note that these three secrets $(\mathrm{CHTS}_0, \mathrm{SHTS}_0, \mathrm{dHS}_0)$ are computed in a manner nearly identical to that of their similarly named counterparts in the modified TLS 1.3 protocol (see Fig. 5). The $\mathsf{O}^{\mathrm{Dec}}$ oracle takes a ciphertext (different from the challenge ciphertext)

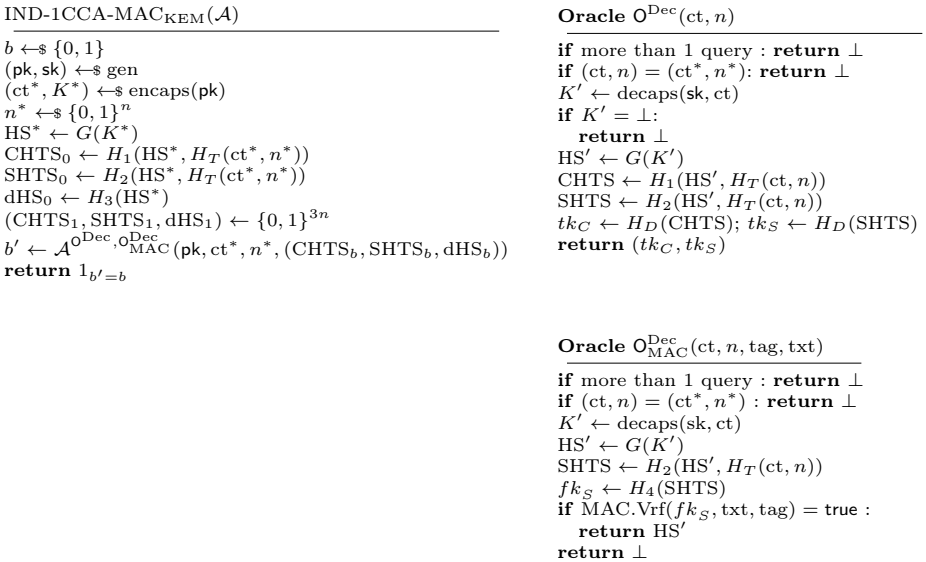

Fig. 4: IND-1CCA-MAC game

and serves as a decapsulation oracle, implementing a key schedule similar to that used in TLS to process the decapsulated key. Ultimately, $\mathsf{O}^{\mathrm{Dec}}$ returns two secrets: tk_C and tk_S. The $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ oracle takes a ciphertext (different from the challenge ciphertext), a tag, and some message txt. The ciphertext is then decrypted to recover a secret HS', which is then subjected to a key schedule to generate a MAC key fk_S. Finally, the oracle verifies whether the tag constitutes a valid MAC on the txt with the key fk_S. If the tag is valid, it returns HS'. Otherwise, it returns an error $\perp$.

Definition 8 (IND-1CCA-MAC*). *The security definition is exactly the same as* IND-1CCA-MAC*, except that it restricts the adversary $\mathcal{A}$ to make only the first query to $\mathsf{O}^{\mathrm{Dec}}$, and subsequent query to $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ must have inputs $(\mathrm{ct}, n, \mathrm{tag}, \mathrm{txt})$ consistent with those in $\mathsf{O}^{\mathrm{Dec}}$. Specifically, the input components (ct, n) for both $\mathsf{O}^{\mathrm{Dec}}$ and $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ must be the same.*

2.3 TLS1.3 protocol

The Transport Layer Security (TLS) protocol is one of the most widely deployed cryptographic protocols in practice. In this paper, we focus on the TLS 1.3 handshake protocol. We refer the reader to [17] for a detailed introduction to TLS 1.3 handshake protocol. Additionally, we recall the notion of the MultiStage security model [17, 22] in Appendix B. We present the (full 1-RTT) handshake of TLS 1.3 with the DH component substituted by a KEM, as shown in Fig. 5. Below, we explicitly demonstrate the correspondence between TLS

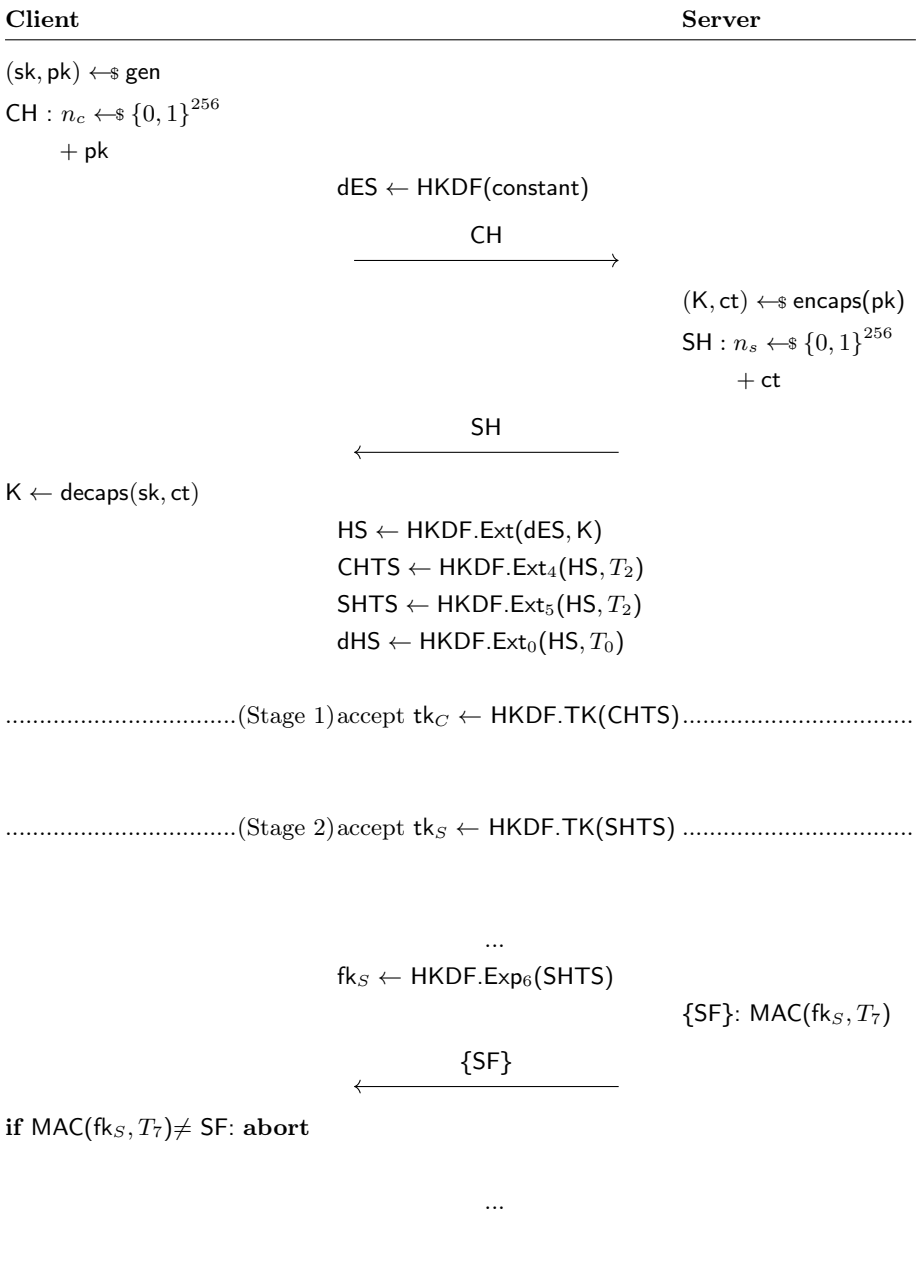

Fig. 5: TLS 1.3 handshake with KEM [22]. {...} denotes a message encrypted with the session traffic key tk_S. T_i denotes the hash of the transcript up to the i-th message. For simplicity, the CH (resp. SH) message captures both the ClientHello and ClientKeyShare (resp. ServerHello and ServerKeyShare). Only the relevant steps for the proof are depicted. Keys in the remaining stages (3-6, not shown) are all derived from the Diffie-Hellman secret (dHS).

1.3 and the IND-1CCA-MAC/IND-1CCA-MAC* game. First, based on $T_2 = H(\mathrm{CH}, \mathrm{SH}) = H(n_c, \mathsf{pk}, n_s, \mathrm{ct})$, we can rewrite $\mathrm{CHTS} = \mathrm{HKDF.Ext}_4(\mathrm{HS}, T_2)$ as $\mathrm{CHTS} = H_j(\mathrm{HS}, H_T(n_s, \mathrm{ct})), j \in \{1, 2\}$, where H_j and H_T are both modeled as random oracles. Here, we omit the public key pk and the client nonce n_c because these values are not important for the proof of Theorem 6, game $G_{B.2}$. Similarly, since dES is constant, one can represent $\mathrm{HKDF.Ext}(\mathrm{dES}, \cdot)$ as $G(\cdot)$, $\mathrm{HKDF.Ext}_0(T_0, \cdot)$ as $H_3(\cdot)$, $\mathrm{HKDF.Exp}_6(\cdot)$ as $H_4(\cdot)$, and $\mathrm{HKDF.TK}(\cdot)$ as $H_D(\cdot)$, where G, H_3, H_4, and H_D are all modeled as random oracles. This rewrite clarifies how these key steps in TLS 1.3 correspond precisely with the IND-1CCA-MAC/IND-1CCA-MAC* game.

3 CPA-secure KEMs are sufficient for TLS 1.3 in the ROM/QROM

In this chapter, we first prove OW-CPA/IND-CPA/D-OW-CPA KEMs are IND-1CCA-MAC* with an EUF-0T secure MAC in the ROM and QROM. We then establish the security proof of TLS 1.3 from IND-1CCA-MAC* in the standard model. By integrating these results, we establish a comprehensive security proof for TLS 1.3 from OW-CPA/IND-CPA/D-OW-CPA KEMs in the ROM and QROM. Our analysis yields significantly tighter bounds than those presented in [22] in the ROM and is the first security proof for TLS 1.3 from CPA-secure KEMs in the QROM.

3.1 OW-CPA/IND-CPA/D-OW-CPA KEMs imply IND-1CCA-MAC/IND-1CCA-MAC* in the ROM

In this section, we demonstrate that OW-CPA/IND-CPA KEMs are IND-1CCA-MAC/IND-1CCA-MAC* secure with an EUF-0T secure MAC in the ROM and D-OW-CPA KEMs are also IND-1CCA-MAC/IND-1CCA-MAC* secure with an EUF-0T secure MAC in the ROM with tight reduction. More precisely, the KDFs G, H_1, H_2, H_3, H_4, and H_D, and the hash function H_T in the IND-1CCA-MAC/IND-1CCA-MAC* games are assumed to be random oracles.

Theorem 1. *Let* $\mathrm{KEM} = (\mathrm{gen}, \mathrm{encaps}, \mathrm{decaps})$ *be a KEM. Let the KDFs and the hash function in the* IND-1CCA-MAC *game be modeled as random oracles. Then, for any PPT adversary* $\mathcal{A}$ *making at most* q_G, q_{H_1}, q_{H_2}, q_{H_3}, q_{H_4}, q_{H_D}, q_{H_T} *queries to* G, H_1, H_2, H_3, H_4, H_D, H_T *respectively, there exists an OW-CPA adversary* $\mathcal{C}$*, an IND-CPA adversary* $\mathcal{D}$*, and an EUF-0T adversary* $\mathcal{B}$ *such that*

$$\begin{aligned}\mathrm{Adv}^{\text{IND-1CCA-MAC}}_{\mathrm{KEM}}(\mathcal{A}) \leq & 2q_G(q_{H_2}+1)(q_{H_1}+q_{H_2}+1) \cdot \mathrm{Adv}^{\text{OW-CPA}}_{\mathrm{KEM}}(\mathcal{C}) \\ & + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}) + \frac{q_{H_1}+2q_{H_2}+q_{H_3}+q_{H_D}+q_{H_T}+6}{2^n}\end{aligned}$$

$$\begin{aligned}\mathrm{Adv}^{\text{IND-1CCA-MAC}}_{\text{KEM}}(\mathcal{A}) \leq & 4(q_{H_2}+1)(q_{H_1}+q_{H_2}+1)\cdot \mathrm{Adv}^{\text{IND-CPA}}_{\text{KEM}}(\mathcal{D}) \\ & + \mathrm{Adv}^{\text{EUF-0T}}_{\text{MAC}}(\mathcal{B}) + \frac{q_{H_1}+2q_{H_2}+q_{H_3}+q_{H_D}+q_{H_T}+6}{2^n} \\ & + \frac{4q_G(q_{H_2}+1)(q_{H_1}+q_{H_2}+1)}{|K|}.\end{aligned}$$

where $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$ have approximately the same running time as $\mathcal{A}$.

Proof. Let $\mathcal{A}$ be an adversary against the IND-1CCA-MAC security of KEM, issuing (exactly) one classical query to $\mathrm{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ and one classical query to $\mathrm{O}^{\mathrm{Dec}}$ (by introducing a dummy query if necessary). We proceed with a sequence of games, which are given in detail in Fig. 6, 7.

GAME G_0. This is the original IND-1CCA-MAC game. From now on, we assume w.l.o.g. that each query to ROs is unique (i.e., they never repeat). Thus, $|\Pr[G^{\mathcal{A}}_0 \Rightarrow 1] - 1/2| = \mathrm{Adv}^{\text{IND-1CCA-MAC}}_{\text{KEM}}(\mathcal{A})$.

GAMES $G_0 - G_8$

$b \leftarrow\$ \{0,1\}, (\mathsf{pk},\mathsf{sk}) \leftarrow \mathrm{gen}, G, H_{k\in\{1-4,T,D\}} \leftarrow\$ \Omega_G, \Omega_{H_i}, H_{12} \leftarrow\$ \Omega_{H_{12}}$

$(\mathrm{ct}^*, K^*) \leftarrow\$ \mathrm{encaps}(\mathsf{pk}), (n^*, \overline{chts}, \overline{shts}) \leftarrow\$ \{0,1\}^{3n}$

$guess \leftarrow\$ \{0,1\}, i \leftarrow\$ [q_{H_1}+q_{H_2}], j \leftarrow\$ [q_{H_2}]$

$\mathrm{HS}^* \leftarrow G(K^*), \mathrm{dHS}_0 \leftarrow H_3(\mathrm{HS}^*)$

$\mathrm{CHTS}_0 \leftarrow H_1(\mathrm{HS}^*, H_T(\mathrm{ct}^*, n^*)), \mathrm{SHTS}_0 \leftarrow H_2(\mathrm{HS}^*, H_T(\mathrm{ct}^*, n^*))$

$(\mathrm{CHTS}_0, \mathrm{SHTS}_0, \mathrm{dHS}_0) \leftarrow\$ \{0,1\}^{3n}$ //G_6-

$(\mathrm{CHTS}_1, \mathrm{SHTS}_1, \mathrm{dHS}_1) \leftarrow\$ \{0,1\}^{3n}$

$b' \leftarrow \mathcal{A}^{G, H_i, \mathsf{O}^{\mathrm{Dec}}, \mathrm{O}^{\mathrm{Dec}}_{\mathrm{MAC}}}(\mathsf{pk}, \mathrm{ct}^*, n^*, (\mathrm{CHTS}_b, \mathrm{SHTS}_b, \mathrm{dHS}_b))$

if $\exists (\mathrm{ct}, n) \neq (\mathrm{ct}^*, n^*), H_T(\mathrm{ct}, n) = H_T(\mathrm{ct}^*, n^*)$: **abort** //G_1-

if $\mathcal{A}$ queries $H_{k_1}(\mathrm{HS}^*, H_T(\mathrm{ct}^*, n^*))$, $k_1 \in \{1,2\}$ or $H_3(\mathrm{HS}^*)$: //QUERY

  if $\mathcal{A}$ did not query $G(K^*)$: **abort** //G_5-

return $1_{b=b'}$

$H_{k_1}(\mathrm{HS}, t)$ //G_4-

$(\mathrm{CHTS}, \mathrm{SHTS}) = H_{12}(\mathrm{HS}, t)$

if $k_1 = 1$ **return** CHTS

else return SHTS

$H_{12}(\mathrm{HS}, t)$ //G_7-

// define $\mathrm{HS}_{l+1} = \mathrm{HS}$

if $l = i$ **return** $\overline{chts}, \overline{shts}$

$l = l + 1$

return $H_{12}(\mathrm{HS}, t)$

Fig. 6: Games for the proof of Thm 1

GAME G_1: In game G_1, we abort if there exists $(\mathrm{ct}, n) \neq (\mathrm{ct}^*, n^*)$ such that $H_T(\mathrm{ct}, n) = H_T(\mathrm{ct}^*, n^*)$. Since there are at most $q_{H_T} + 4$ queries to H_T in the game (including two additional implicit queries to H_T in both $\mathsf{O}^{\mathrm{Dec}}$ and $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$), the probability of $\mathcal{A}$ finding such (ct, n) given by (ct^*, n^*) is less than $(q_{H_T} + 4)/2^n$.

$$|\Pr[G_0^{\mathcal{A}} \Rightarrow 1] - \Pr[G_1^{\mathcal{A}} \Rightarrow 1]| \leq \frac{q_{H_T} + 4}{2^n}.$$

GAME G_2. In game G_2, we abort whenever the MAC verification succeeds on the query $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}(\mathrm{ct}_2, n_2, \mathrm{tag}, \mathrm{txt})$ but $fk_S := H_4(\mathrm{SHTS})$ was never queried before the query of $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}(\mathrm{ct}_2, n_2, \mathrm{tag}, \mathrm{txt})$, where $\mathrm{SHTS} := H_2(G(K), H_T(\mathrm{ct}_2, n_2))$ and $K := \mathrm{Decaps}(\mathsf{sk}, \mathrm{ct}_2)$. If this is the case, it means the MAC key $fk_S := H_4(\mathrm{SHTS})$ is indistinguishable from a random value for $\mathcal{A}$ when it queries the $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}(\mathrm{ct}_2, n_2, \mathrm{tag}, \mathrm{txt})$ oracle, but it manages to forge a valid tag. Therefore, we can build an adversary $\mathcal{B}$ that breaks MAC unforgeability. More formally, $\mathcal{B}$ samples a pair of keys $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathrm{gen}$ for KEM, generates a valid input for $\mathcal{A}$ and simulates the $\mathsf{O}^{\mathrm{Dec}}$ oracle with the secret key. Then, when $\mathcal{A}$ query $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}(\mathrm{ct}_2, n_2, \mathrm{tag}, \mathrm{txt})$, $\mathcal{B}$ outputs $(\mathrm{txt}, \mathrm{tag})$ as a forgery.

We also abort if the value SHTS computed in the $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ oracle is such that $\mathrm{SHTS} = \mathrm{SHTS}_b$. Since G_1 and G_2 return $\perp$ when $H_T(\mathrm{ct}_2, n_2) = H_T(\mathrm{ct}^*, n^*)$, the event $\mathrm{SHTS} = \mathrm{SHTS}_b = H_2(., H_T(\mathrm{ct}^*, n^*))$ occurs implies the adversary that given SHTS_b finds $H_2(., H_T(\mathrm{ct}_2, n_2)) = \mathrm{SHTS}_b$ such that $H_T(\mathrm{ct}_2, \mathrm{n}_2) \neq H_T(\mathrm{ct}^*, n^*)$. Note that there are at most $q_{H_2} + 1$ queries to H_2 (including an implicit query in $\mathsf{O}^{\mathrm{Dec}}$), the probability of the event $\mathrm{SHTS} = \mathrm{SHTS}_b$ occurring is at most $(q_{H_2} + 1)/2^n$. Therefore, we have

$$|\Pr[G_1^{\mathcal{A}} \Rightarrow 1] - \Pr[G_2^{\mathcal{A}} \Rightarrow 1]| \leq \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}) + \frac{q_{H_2} + 1}{2^n}.$$

GAME G_3. In game G_3, we abort whenever the MAC verification succeeds on the query $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}(\mathrm{ct}_2, n_2, \mathrm{tag}, \mathrm{txt})$ but $H_2(G(K), H_T(\mathrm{ct}_2, n_2))$ was never queried before the query of $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}(\mathrm{ct}_2, n_2, \mathrm{tag}, \mathrm{txt})$, where $K := \mathrm{decaps}(\mathsf{sk}, \mathrm{ct}_2)$. By the previous game, it means that the adversary queried $\mathrm{SHTS} := H_2(G(K), H_T(\mathrm{ct}_2, n_2))$ to H_4 without having queried $H_2(G(K), H_T(\mathrm{ct}_2, n_2))$ beforehand. Let us analyze what information $\mathcal{A}$ has about $\mathrm{SHTS} \neq \mathrm{SHTS}_b$ if it did not query $H_2(G(K), H_T(\mathrm{ct}_2, n_2))$. Note that the only potential "leakage" is from $\mathsf{O}^{\mathrm{Dec}}$ that returns $tk_S := H_D(\mathrm{SHTS})$, where H_D is a RO perfectly hiding SHTS. If G_3 aborts, we can construct an adversary $\mathcal{A}$ who is capable of recovering SHTS given the random oracle output $H_D(\mathrm{SHTS})$. The best strategy for $\mathcal{A}$ to recover SHTS is to query random values $x \in \{0,1\}^n$ to H_D until it finds x such that $H_D(x) = tk_S$, or randomly guess the value of SHTS. Thus, the advantage of $\mathcal{A}$ is at most $\frac{q_{H_D}+1}{2^n}$. Hence, we have

$$|\Pr[G_2^{\mathcal{A}} \Rightarrow 1] - \Pr[G_3^{\mathcal{A}} \Rightarrow 1]| \leq \frac{q_{H_D} + 1}{2^n}.$$

GAME G_4. In game G_4, we use a new random oracle $H_{12}(\mathrm{HS}, t)$ to simulate both H_1 and H_2. Let $H_{12}(\mathrm{HS}, t) := (H_1(\mathrm{HS}, t), H_2(\mathrm{HS}, t))$. Here, H_{12} is a random oracle maintained internally by the challenger. When adversary $\mathcal{A}$ queries $H_1(\mathrm{HS}, t)$, the challenger only needs to query $H_{12}(\mathrm{HS}, t)$ and then output the first component. Similarly, when adversary $\mathcal{A}$ queries $H_2(\mathrm{HS}, t)$, the challenger only needs to query $H_{12}(\mathrm{HS}, t)$ and then output the second component. Consequently, the total number of queries to H_{12} is at most $q_{H_1} + q_{H_2}$. Therefore, this G_4 is consistent with G_3 from $\mathcal{A}$'s view. In other words, we have

$$\Pr[G_3^{\mathcal{A}} \Rightarrow 1] = \Pr[G_4^{\mathcal{A}} \Rightarrow 1].$$

$\mathsf{O}^{\mathrm{Dec}}(\mathrm{ct}_1, n_1)$:

1 : **if** more than 1 query :
2 : **return** $\perp$
3 : **if** $(\mathrm{ct}_1, n_1) = (\mathrm{ct}^*, n^*)$:
4 : **return** $\perp$
5 : **if** $guess = 0$: // G_7-
6 : **return** $\perp$
7 : **return** $H_D(\overline{chts}), H_D(\overline{shts})$
8 : // G_7-
9 : $K' \leftarrow \mathrm{decaps}(\mathsf{sk}, \mathrm{ct}_1)$
10 : **if** $K' = \perp$: **return** $\perp$
11 : $\mathrm{HS}' \leftarrow G(K')$
12 : $\mathrm{CHTS} \leftarrow H_1(\mathrm{HS}', H_T(\mathrm{ct}_1, n_1))$
13 : $\mathrm{SHTS} \leftarrow H_2(\mathrm{HS}', H_T(\mathrm{ct}_1, n_1))$
14 : $tk_C \leftarrow H_D(\mathrm{CHTS})$
15 : $tk_S \leftarrow H_D(\mathrm{SHTS})$
16 : **return** tk_C, tk_S

$\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}(\mathrm{ct}_2, n_2, \mathrm{tag}, \mathrm{txt})$

1 : **if** more than 1 query : **return** $\perp$
2 : **if** $(\mathrm{ct}_2, n_2) = (\mathrm{ct}^*, n^*)$: **return** $\perp$
3 : **if** $(\mathrm{ct}_2, n_2) = (\mathrm{ct}_1, n_1)$: // G_7-
4 : **if** $guess = 0$: **return** $\perp$
5 : $fk_S \leftarrow H_4(\overline{shts})$
6 : **if** MAC.Vrf(fk_S,txt,tag) = true:
7 : **return** HS_{i+1}
8 : **return** $\perp$
9 : **if** $(\mathrm{ct}_2, n_2) \neq (\mathrm{ct}_1, n_1)$ // G_8-
10 : $\vee \mathcal{A}$ did not query $\mathsf{O}^{\mathrm{Dec}}$:
11 : **if** $j = q_{H_2}$: **return** $\perp$
12 : **else return** HS_{j+1}
13 : $K' \leftarrow \mathrm{decaps}(\mathsf{sk}, \mathrm{ct}_2)$, $\mathrm{HS}' \leftarrow G(K')$
14 : $\mathrm{SHTS} \leftarrow H_2(\mathrm{HS}', H_T(\mathrm{ct}_2, n_2))$
15 : $fk_S \leftarrow H_4(\mathrm{SHTS})$
16 : **if** $\mathrm{SHTS} = \mathrm{SHTS}_b$: **abort** // G_2-
17 : **if** MAC.Vrf(fk_S,txt,tag) = true:
18 : **if** $\mathcal{A}$ did not query $H_4(\mathrm{SHTS})$:
19 : **abort** // G_2-
20 : **if** $\mathcal{A}$ did not query
21 : $H_2(\mathrm{HS}', H_T(\mathrm{ct}_2, n_2))$:
22 : **abort** // G_3-
23 : **return** HS'
24 : **return** $\perp$

Fig. 7: Games for the proof of Thm 1

GAME G_5. In game G_5, we abort whenever the adversary did not query $G(K^*)$ (which is equal to HS^*) but queried $H_1(\mathrm{HS}^*, H_T(\mathrm{ct}^*, n^*))$, $H_2(\mathrm{HS}^*, H_T(\mathrm{ct}^*, n^*))$, or $H_3(\mathrm{HS}^*)$. Note that the $\mathsf{O}^{\mathrm{Dec}}$ and $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ never query $H_1(\mathrm{HS}^*, H_T(\mathrm{ct}^*, n^*))$, $H_2(\mathrm{HS}^*, H_T(\mathrm{ct}^*, n^*))$, or $H_3(\mathrm{HS}^*)$ and the challenge values given to $\mathcal{A}$ are either perfectly random or completely hide HS^*. Similarly, the $\mathsf{O}^{\mathrm{Dec}}$ oracle also completely hides HS^*. According to the definition of the previous game, the $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}(\mathrm{ct}_2, n_2, \mathrm{tag}, \mathrm{txt})$ oracle returns HS^* if and only if the corresponding $(\mathrm{HS}^*, H_T(\mathrm{ct}_2, n_2))$ was queried to H_2 before, where $\mathrm{HS}^* = G(K')$ and $K' = \mathrm{decaps}(\mathsf{sk}, \mathrm{ct}_2)$. That is, the adversary has queried $H_2(\mathrm{HS}^*, H_T(\mathrm{ct}_2, n_2))$ before the $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ oracle, which means the adversary already knows HS^*. Therefore, HS^* is uniformly random and independent from $\mathcal{A}$'s view, and the probability that $\mathcal{A}$ queries HS^* to H_1, H_2, or H_3 is upper bounded by $\frac{q_{H_1}+q_{H_2}+q_{H_3}}{2^n}$. Hence, we have

$$|\Pr[G_4^{\mathcal{A}} \Rightarrow 1] - \Pr[G_5^{\mathcal{A}} \Rightarrow 1]| \leq \frac{q_{H_1} + q_{H_2} + q_{H_3}}{2^n}.$$

GAME G_6. In game G_6, $(\mathrm{CHTS}_0, \mathrm{SHTS}_0, \mathrm{dHS}_0)$ is replaced by $(\mathrm{CHTS}_0, \mathrm{SHTS}_0, \mathrm{dHS}_0) \leftarrow\$ \{0,1\}^{3n}$. Define QUERY as the event that $H_1(\mathrm{HS}^*, H_T(\mathrm{ct}^*, n^*))$, $H_2(\mathrm{HS}^*, H_T(\mathrm{ct}^*, n^*))$, or $H_3(\mathrm{HS}^*)$ is queried by the adversary. Then, G_6 is identical to G_5 in $\mathcal{A}$'s view unless the event QUERY happens. Thus, we have

$$\left|\Pr[G_5^{\mathcal{A}} \Rightarrow 1] - \Pr[G_6^{\mathcal{A}} \Rightarrow 1]\right| \leq \Pr[\mathrm{QUERY} : G_6].$$

It is evident that in game G_6, the bit b is independent of adversary $\mathcal{A}$'s view. Therefore, we have

$$\Pr[G_6^{\mathcal{A}} \Rightarrow 1] = \frac{1}{2}.$$

GAME G_7. In game G_7, the challenger can simulate the $\mathsf{O}^{\mathrm{Dec}}(\mathrm{ct}_1, n_1)$ oracle without the secret key. Initially, we randomly sample $guess \leftarrow\$ \{0,1\}$ to guess whether $K' = \mathrm{decaps}(\mathsf{sk}, \mathrm{ct}_1)$ equals to $\perp$. If $guess = 0$, in this case, we guess $K' = \perp$, so we just return $\perp$ in $\mathsf{O}^{\mathrm{Dec}}$. Otherwise, if $guess = 1$, we will make two further changes in this game. First, we modify the Dec oracle and replace $\mathrm{CHTS} := H_1(\mathrm{HS}', H_T(\mathrm{ct}_1, n_1))$ and $\mathrm{SHTS} := H_2(\mathrm{HS}', H_T(\mathrm{ct}_1, n_1))$ with $\mathrm{CHTS} = \overline{chts}$ and $\mathrm{SHTS} = \overline{shts}$, where $\overline{chts}$ and $\overline{shts}$ are randomly chosen from $\{0,1\}^n$ and $\mathrm{HS}' = G(K')$. Second, we reprogram the random oracle H_{12} conditionally on a uniform i over $[q_{H_1} + q_{H_2}]$. In other words, on the $i+1$-th query, we reprogram H_{12} to return $(\overline{chts}, \overline{shts})$, while keeping all other queries unchanged. Let $(i^* + 1)$ denote the number of first queries to H_{12} with $(\mathrm{HS}', H_T(\mathrm{ct}_1, n_1))$, where $i^* \in [q_{H_1} + q_{H_2} - 1]$. We also denote $i^* = q_{H_1} + q_{H_2}$ as the event that H_{12} makes no queries with $(\mathrm{HS}', H_T(\mathrm{ct}_1, n_1))$.

Moreover, if the $(\mathrm{ct}_2, n_2, ., .)$ input to $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ is equal to the (ct_1, n_1) input to $\mathsf{O}^{\mathrm{Dec}}$, that is, $(\mathrm{ct}_1, n_1) = (\mathrm{ct}_2, n_2)$. We change $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ as follows: In the case of $guess = 0$, we return $\perp$ in $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$. Otherwise, we compute $fk_S = H_4(\overline{shts})$

and verify if MAC.Vrf(fk_S, txt, tag) is correct. If the verification returns true, we only need to extract HS_{i+1} from the $i+1$-th query to H_{12}, and then output HS_{i+1}. According to the previous game, when MAC.Vrf(fk_S, txt, tag) is correct, we can conclude that the adversary has already queried the corresponding H_2, which means the $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ oracle query must have occurred after the $i+1$-th H_{12} query so that we can extract the corresponding HS_{i+1}. If the verification is not true, we output $\perp$.

Note that G_7 has the same distribution as G_6 in $\mathcal{A}$'s view when the event $i^* = i$ occurs and the *guess* is correct. Thus, we have

$$\Pr[\mathrm{QUERY} : G_6] \leq 2(q_{H_1} + q_{H_2} + 1) \Pr[\mathrm{QUERY} : G_7].$$

GAME G_8. In game G_8, the challenger can simulate $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}(\mathrm{ct}_2, n_2, \mathrm{tag}, \mathrm{txt})$ oracle without the secret key. We modify the $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ oracle as follows: If the $(\mathrm{ct}_2, n_2, ., .)$ input to $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ is not equal to the (ct_1, n_1) input to $\mathsf{O}^{\mathrm{Dec}}$ or $\mathcal{A}$ did not query $\mathsf{O}^{\mathrm{Dec}}$ before. According to the definition of the previous game, this oracle returns something other than $\perp$ if and only if the corresponding $(\mathrm{HS}', H_T(\mathrm{ct}_2, n_2))$ was queried to H_2 before, where $\mathrm{HS}' = G(K')$ and $K' = \mathrm{decaps}(\mathsf{sk}, \mathrm{ct}_2)$. Therefore, one can randomly choose j over $[q_{H_2}]$ and guess whether $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ outputs $\perp$ (if $j = q_{H_2}$) or corresponding HS' is in the $j+1$-th [8] query and return HS_{j+1}. Therefore, the simulation is such that when $j = q_{H_2}$, output $\perp$. Otherwise, we extract HS_{j+1} from the $j+1$-th query to H_2 (simulated by H_{12}), and then output HS_{j+1}. Overall, the simulation works with probability $\frac{1}{(q_{H_2}+1)}$. Thus, we have

$$\Pr[\mathrm{QUERY} : G_7] \leq (q_{H_2} + 1) \Pr[\mathrm{QUERY} : G_8].$$

Note that if QUERY happens, K^* will be in the G-list of queries made by $\mathcal{A}$. Let $(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$$ gen, $(K^*, \mathrm{ct}^*) \leftarrow\!\!\$$ encaps(pk). Then, we construct an adversary $\mathcal{C}'(\mathsf{pk}, \mathrm{ct}^*)$ samples $n^*, \overline{chts}, \overline{shts}, guess, i, j$ as in G_8 and $\mathrm{CHTS}^*, \mathrm{SHTS}^*, \mathrm{dHS}^* \leftarrow\!\!\$ \{0,1\}^{3n}$. Then $\mathcal{C}$ picks five q_{H_k}-wise ($k \in \{12, 3, 4, H, T\}$) independent functions and a q_G-wise independent functions (indistinguishable from a random function for a $q_{H_k}(q_G)$-query adversary according to [38]) and runs $\mathcal{A}^{G, H_i, \mathsf{O}^{\mathrm{Dec}}, \mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}}(pk, \mathrm{ct}^*, n^*, \mathrm{CHTS}^*, \mathrm{SHTS}^*, \mathrm{dHS}^*)$ as in game G_8 and returns $\mathcal{A}$'s G-query list G-List.

Now, we can construct an adversary $\mathcal{C}$ against the OW-CPA security of the underlying KEM. $\mathcal{C}$ runs $\mathcal{C}'$ and randomly selects one message in the G-List as a return. Then, we have $\mathrm{Adv}^{\text{OW-CPA}}_{\mathrm{KEM}}(\mathcal{C}) \geq \frac{1}{q_G} \Pr[\mathrm{Query} : G_8]$. Therefore, we have

$$\begin{aligned}\mathrm{Adv}^{\text{IND-1CCA-MAC}}_{\mathrm{KEM}}(\mathcal{A}) \leq & 2q_G(q_{H_2}+1)(q_{H_1}+q_{H_2}+1) \cdot \mathrm{Adv}^{\text{OW-CPA}}_{\mathrm{KEM}}(\mathcal{C}) \\ & + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}) + \frac{q_{H_1} + 2q_{H_2} + q_{H_3} + q_{H_D} + q_{H_T} + 6}{2^n}.\end{aligned}$$

[8] Here, we exclude the H_{12} query induced by the H_1 query from the adversary.

Right now, we consider the case of the IND-CPA KEM. Specifically, the IND-CPA challenger generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\$$ gen, $(\mathrm{ct}^*, K^*) \leftarrow\!\$$ encaps(pk), and $b' \leftarrow\!\$ \{0,1\}$. When $b' = 0$, define $K^*_{b'} = K^*$, and when $b' = 1$, define $K^*_{b'} \leftarrow\!\$ K$. Finally, $\mathcal{D}$ needs to guess the value of b' after receiving $(\mathsf{pk}, K^*_{b'}, \mathrm{ct}^*)$ from the challenger.

Then, $\mathcal{D}$ runs $\mathcal{C}'(\mathsf{pk}, \mathrm{ct}^*)$ to get $\mathcal{A}$'s G-List. Let BADG be the event that the G-List contains K^*_1. Since K^*_1 is uniformly random and independent from $\mathcal{A}$'s view, the probability that adversary $\mathcal{A}$ queries $G(K^*_1)$ is at most $\frac{q_G}{|K|}$. For the remainder of the proof, we assume BADG did not happen. If QUERY happens, this means adversary $\mathcal{A}$ queried the random oracle G on K^*_0. In this case, if $\mathcal{D}$ obtains $K^*_{b'}$ in the G-List, it directly outputs $b = 0$; otherwise, it outputs $b = 1$. If QUERY does not happen, $\mathcal{D}$ uniformly randomly guesses the value of b', i.e., it outputs $b \leftarrow\!\$ \{0,1\}$. Thus, we have

$$\begin{aligned}
\mathrm{Adv}^{\text{IND-CPA}}_{\text{KEM}}(\mathcal{D}) + \frac{q_G}{|K|} &\geq |\Pr[b' = b] - \frac{1}{2}| \\
&= |\Pr[\text{QUERY}:G_8] + \frac{1}{2}\Pr[\neg\text{QUERY}:G_8] - \frac{1}{2}| \\
&= \frac{1}{2}\Pr[\text{QUERY}:G_8].
\end{aligned}$$

Putting the bounds together, we have

$$\begin{aligned}
\mathrm{Adv}^{\text{IND-1CCA-MAC}}_{\text{KEM}}(\mathcal{A}) \leq& 4(q_{H_2}+1)(q_{H_1}+q_{H_2}+1)\cdot \mathrm{Adv}^{\text{IND-CPA}}_{\text{KEM}}(\mathcal{D}) \\
&+ \mathrm{Adv}^{\text{EUF-0T}}_{\text{MAC}}(\mathcal{B}) + \frac{q_{H_1}+2q_{H_2}+q_{H_3}+q_{H_D}+q_{H_T}+6}{2^n} \\
&+ \frac{4q_G(q_{H_2}+1)(q_{H_1}+q_{H_2}+1)}{|K|}.
\end{aligned}$$

□

Theorem 2. *Let* KEM = (gen, encaps, decaps) *be a KEM. Let the KDFs and the hash function in the* IND-1CCA-MAC* *game be modeled as random oracles. Then, for any PPT adversary* $\mathcal{A}$ *making at most* q_G, q_{H_1}, q_{H_2}, q_{H_3}, q_{H_4}, q_{H_D}, q_{H_T} *queries to* G, H_1, H_2, H_3, H_4, H_D, H_T *respectively, there exists a* OW-CPA *adversary* $\mathcal{C}$ *, an* IND-CPA *adversary* $\mathcal{D}$ *and an EUF-0T adversary* $\mathcal{B}$ *such that*

$$\begin{aligned}
\mathrm{Adv}^{\text{IND-1CCA-MAC}^*}_{\text{KEM}}(\mathcal{A}) \leq& 2q_G(q_{H_1}+q_{H_2}+1)\cdot \mathrm{Adv}^{\text{OW-CPA}}_{\text{KEM}}(\mathcal{C}) \\
&+ \mathrm{Adv}^{\text{EUF-0T}}_{\text{MAC}}(\mathcal{B}) + \frac{q_{H_1}+2q_{H_2}+q_{H_3}+q_{H_D}+q_{H_T}+6}{2^n}
\end{aligned}$$

$$\begin{aligned}
\mathrm{Adv}^{\text{IND-1CCA-MAC}^*}_{\text{KEM}}(\mathcal{A}) \leq& 4(q_{H_1}+q_{H_2}+1)\cdot \mathrm{Adv}^{\text{IND-CPA}}_{\text{KEM}}(\mathcal{D}) \\
&+ \mathrm{Adv}^{\text{EUF-0T}}_{\text{MAC}}(\mathcal{B}) + \frac{q_{H_1}+2q_{H_2}+q_{H_3}+q_{H_D}+q_{H_T}+6}{2^n} \\
&+ \frac{4q_G(q_{H_2}+1)(q_{H_1}+q_{H_2}+1)}{|K|}.
\end{aligned}$$

Proof. It is easy to see that, except for G_8, the proof is the same as the proof of Theorem 1. The G_8 is redundant because IND-1CCA-MAC* require $(\mathrm{ct}_1, n_1) = (\mathrm{ct}_2, n_2)$ and the query $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ is subsequent to the query $\mathsf{O}^{\mathrm{Dec}}$. Thus, in this proof, we can simply define G_8 to be identical to G_7 to maintain consistency with Theorem 1. Hence, we have $\Pr[\mathrm{QUERY} : G_7] = \Pr[\mathrm{QUERY} : G_8]$.

□

Theorem 3. *Let* $\mathrm{KEM} = (\mathrm{gen}, \mathrm{encaps}, \mathrm{decaps}) = S(\mathrm{DPKE}(\mathrm{gen}', \mathrm{enc}', \mathrm{dec}'))$ [9] *be a DKEM and the underlying* δ*-correctness DPKE is rigid. Let the KDFs and the hash function in the* IND-1CCA-MAC *game be modeled as random oracles. Then, for any PPT adversary* $\mathcal{A}$ *making at most* q_G*,* q_{H_1}*,* q_{H_2}*,* q_{H_3}*,* q_{H_4}*,* q_{H_D}*,* q_{H_T} *queries to* G*,* H_1*,* H_2*,* H_3*,* H_4*,* H_D*,* H_T *respectively, there exists an OW-CPA adversary* $\mathcal{C}$ *and an EUF-0T adversary* $\mathcal{B}$ *such that*

$$\begin{aligned}\mathrm{Adv}^{\text{IND-1CCA-MAC}}_{\mathrm{KEM}}(\mathcal{A}) \leq & 4\mathrm{Adv}^{\text{OW-CPA}}_{\mathrm{DKEM}}(\mathcal{C}) + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}) \\ & + \frac{q_{H_1} + 2q_{H_2} + q_{H_3} + q_{H_D} + q_{H_T} + 6}{2^n} + \delta\end{aligned}$$

where $\mathcal{C}$ *have approximately the same running time as* $\mathcal{A}$*.*

Proof. Let $\mathcal{A}$ be an adversary against the IND-1CCA-MAC security of KEM, issuing one classical query to $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ and one classical query to $\mathsf{O}^{\mathrm{Dec}}$ (by introducing a dummy query if necessary). Let $\mathrm{DKEM} = S(\mathrm{DPKE}(\mathrm{gen}', \mathrm{enc}', \mathrm{dec}'))$. In this proof, we utilize the rigid property to simulate the $\mathsf{O}^{\mathrm{Dec}}(\mathrm{ct}_1, n_1)$ and $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}(\mathrm{ct}_2, n_2, \mathrm{tag}, \mathrm{txt})$ tightly. Additionally, we utilize the deterministic property to embed the challenge tightly. Define games $G_0 - G_9$ as in Fig. 8, 9.

GAMES $G_0 - G_3$. Games $G_0 - G_3$ are identical to the $G_0 - G_3$ in Theorem 1.

GAME G_4. In game G_4, we abort whenever the adversary did not query $G(K^*)$ (which is equal to HS^*) but it queried $H_1(\mathrm{HS}^*, H_T(\mathrm{ct}^*, n^*))$, $H_2(\mathrm{HS}^*, H_T(\mathrm{ct}^*, n^*))$ or $H_3(\mathrm{HS}^*)$. Similar to Theorem 1, the probability that $\mathcal{A}$ queries HS^* to H_1, H_2 or H_3 is upper bounded by $\frac{q_{H_1}+q_{H_2}+q_{H_3}}{2^n}$ and hence we have

$$|\Pr[G_3^{\mathcal{A}} \Rightarrow 1] - \Pr[G_4^{\mathcal{A}} \Rightarrow 1]| \leq \frac{q_{H_1} + q_{H_2} + q_{H_3}}{2^n}.$$

GAME G_5. In game G_5, $(\mathrm{CHTS}^*_0, \mathrm{SHTS}^*_0, \mathrm{dHS}^*_0)$ is replaced by $(\mathrm{CHTS}^*_0, \mathrm{SHTS}^*_0, \mathrm{dHS}^*_0) \leftarrow \{0,1\}^{3n}$. Define QUERY as the event that $H_1(\mathrm{HS}^*, H_T(\mathrm{ct}^*, n^*))$, $H_2(\mathrm{HS}^*, H_T(\mathrm{ct}^*, n^*))$, or $H_3(\mathrm{HS}^*)$ is queried by the adversary. Then, G_5 is identical to G_4 in $\mathcal{A}$'s view unless the event QUERY happens. Thus, we have

$$\left|\Pr[G_4^{\mathcal{A}} \Rightarrow 1] - \Pr[G_5^{\mathcal{A}} \Rightarrow 1]\right| \leq \Pr[\mathrm{QUERY} : G_5].$$

[9] Applying the simple CPA transform depicted in Fig. 1 to DPKE.

One can see that in G_5, the bit b is independent of $\mathcal{A}$'s view, thus

$$\Pr[G_5^{\mathcal{A}} \Rightarrow 1] = \frac{1}{2}.$$

GAMES $G_0 - G_9$

$b \leftarrow\$ \{0,1\}, (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathrm{gen}, G, H_{i\in\{1-4,T,D\}} \leftarrow\$ \Omega_G, \Omega_{H_i}$
$n^* \leftarrow\$ \{0,1\}^n, guess, guess' \leftarrow\$ \{0,1\}$
$(\mathrm{ct}^*, K^*) \leftarrow\$ \mathrm{encaps}(\mathsf{pk}), \mathrm{HS}^* \leftarrow G(K^*), \mathrm{dHS}_0 \leftarrow H_3(\mathrm{HS}^*)$
$\mathrm{CHTS}_0 \leftarrow H_1(\mathrm{HS}^*, H_T(\mathrm{ct}^*, n^*)), \mathrm{SHTS}_0 \leftarrow H_2(\mathrm{HS}^*, H_T(\mathrm{ct}^*, n^*))$
$(\mathrm{CHTS}_0, \mathrm{SHTS}_0, \mathrm{dHS}_0) \leftarrow\$ \{0,1\}^{3n}$ $//G_5-$
$(\mathrm{CHTS}_1, \mathrm{SHTS}_1, \mathrm{dHS}_1) \leftarrow\$ \{0,1\}^{3n}$
$b' \leftarrow \mathcal{A}^{G, H_i, \mathsf{O}^{\mathrm{Dec}}, \mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}}(\mathsf{pk}, \mathrm{ct}^*, n^*, (\mathrm{CHTS}_b, \mathrm{SHTS}_b, \mathrm{dHS}_b))$
if $\exists(\mathrm{ct}, n) \neq (\mathrm{ct}^*, n^*), H_T(\mathrm{ct}, n) = H_T(\mathrm{ct}^*, n^*)$: **abort** $//G_1-$
if $\mathcal{A}$ queries $H_k(\mathrm{HS}^*, H_T(\mathrm{ct}^*, n^*)),\ k \in \{1,2\}$ or $H_3(\mathrm{HS}^*)$: //QUERY
  if $\mathcal{A}$ did not query $G(K^*)$: **abort** $//G_4-$
if $\exists(K, \mathrm{HS}) \in \mathcal{L}_G$ s. th. $\mathrm{dec}'(\mathsf{sk}, \mathrm{enc}'(\mathsf{pk}, K)) \neq K$: **abort** $//G_7-$
return $1_{b=b'}$

$\mathsf{O}^{\mathrm{Dec}}(\mathrm{ct}_1, n_1)$

if more than 1 query $\vee$ $(\mathrm{ct}_1, n_1) = (\mathrm{ct}^*, n^*)$: **return** $\perp$
$K' \leftarrow \mathrm{decaps}(\mathsf{sk}, \mathrm{ct}_1)$ $//G_0 - G_7$
if $K' = \perp$: **return** $\perp$ // $G_0 - G_7$
$\mathrm{HS}' \leftarrow G(K')$ // $G_0 - G_7$
if $guess = 0$: **return** $\perp$ // G_8-
if $\exists K'$ s. th. $(K', \mathrm{ct}_1, \mathrm{HS}') \in \mathfrak{L}$: extract HS' // G_8-
else $\mathrm{HS}' \leftarrow\$ \{0,1\}^n$, $\mathfrak{L}_{\mathrm{Dec}} = \{\mathrm{ct}_1, \mathrm{HS}'\}$ // G_8-
$\mathrm{CHTS} \leftarrow H_1(\mathrm{HS}', H_T(\mathrm{ct}_1, n_1))$, $\mathrm{SHTS} \leftarrow H_2(\mathrm{HS}', H_T(\mathrm{ct}_1, n_1))$
$tk_C \leftarrow H_D(\mathrm{CHTS})$, $tk_S \leftarrow H_D(\mathrm{SHTS})$
return tk_C, tk_S

Fig. 8: Games for the proof of Thm 3

GAME G_6. In game G_6, the challenger simulates the random oracle G as follows: When adversary $\mathcal{A}$ queries $G(K)$, return $G(K)$ if it has been previously defined, otherwise randomly select $\mathrm{HS} \leftarrow\$ \{0,1\}^n$ and return it. We also compute $\mathrm{ct} = \mathrm{enc}'(\mathsf{pk}, K)$, and update $\mathfrak{L}_G = \mathfrak{L}_G \cup (K, \mathrm{HS})$, $\mathfrak{L} = \mathfrak{L} \cup (K, \mathrm{ct}, \mathrm{HS})$. We have

$$\Pr[\mathrm{QUERY} : G_5] = \Pr[\mathrm{QUERY} : G_6].$$

$\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}(\mathrm{ct}_2, n_2, \mathrm{tag}, \mathrm{txt})$

1 : **if** more than 1 query $\vee$ $(\mathrm{ct}_2, n_2) = (\mathrm{ct}^*, n^*)$: **return**$\perp$
2 : $K' \leftarrow \mathrm{decaps}(\mathsf{sk}, \mathrm{ct}_2), \mathrm{HS}' \leftarrow G(K')$ $//G_0 - G_8$
3 : **if** $(\mathrm{ct}_2, n_2) = (\mathrm{ct}_1, n_1)$: // G_8-
4 : **if** $guess = 0$: **return** $\perp$
5 : **if** $\exists K'$ s. th. $(K', \mathrm{ct}_1, \mathrm{HS}') \in \mathfrak{L}$: extract HS'
6 : **else** $\mathrm{HS}' \leftarrow \mathfrak{L}_{\mathrm{Dec}} = \{\mathrm{ct}_1, \mathrm{HS}'\}$
7 : **if** $(\mathrm{ct}_2, n_2) \neq (\mathrm{ct}_1, n_1) \vee \mathcal{A}$ did not query $\mathsf{O}^{\mathrm{Dec}}$ before : // G_9
8 : **if** $guess' = 0$: **return** $\perp$
9 : **if** $\exists K'$ s. th. $(K', \mathrm{ct}_2, \mathrm{HS}) \in \mathfrak{L}$: extract HS'
10 : **else** $\mathrm{HS}' \leftarrow\!\!\$\ \{0,1\}^n$, $\mathfrak{L}^{\mathrm{MAC}}_{\mathrm{Dec}} = \{\mathrm{ct}_2, \mathrm{HS}'\}$
11 : $\mathrm{SHTS} \leftarrow H_2(\mathrm{HS}', H_T(\mathrm{ct}, n)), fk_S \leftarrow H_4(\mathrm{SHTS})$
12 : **if** $\mathrm{SHTS} = \mathrm{SHTS}_b$: **abort** // G_2-
13 : **if** MAC.Vrf(fk_S,txt,tag) = true:
14 : **if** $\mathcal{A}$ did not query $H_4(\mathrm{SHTS})$: **abort** // G_2-
15 : **if** $\mathcal{A}$ did not query $H_2(\mathrm{HS}', H_T(\mathrm{ct}_2, n_2))$: **abort** // G_3-
16 : **return** HS'
17 : **return** $\perp$

$G(K)$ // G_6-

1 : **if** $\exists$HS s.th. $(K, \mathrm{HS}) \in \mathfrak{L}_G$: **return** HS
2 : $\mathrm{ct} = \mathrm{enc}'(\mathsf{pk}, K)$, $\mathrm{HS} \leftarrow\!\!\$\ \{0,1\}^n$
3 : **if** $\mathrm{ct} = \mathrm{ct}_1$: HS'$\leftarrow \mathfrak{L}_{\mathrm{Dec}} = \{\mathrm{ct}_1, \mathrm{HS}'\}$, HS=HS$'$ // G_8, G_9
4 : **if** $\mathrm{ct} = \mathrm{ct}_2$: $\mathrm{HS}' \leftarrow \mathfrak{L}^{\mathrm{MAC}}_{\mathrm{Dec}} = \{\mathrm{ct}_2, \mathrm{HS}'\}$, HS=HS$'$ // G_9
5 : $\mathfrak{L}_G = \mathfrak{L}_G \cup \{K, \mathrm{HS}\}, \mathfrak{L} = \mathfrak{L} \cup \{K, \mathrm{ct}, \mathrm{HS}\}$
6 : **return** HS

Fig. 9: Games for the proof of Thm 3

GAME G_7. In game G_7, we define ERO as the event that $\mathcal{L}_G$ contanins an entry (K, HS) with $\mathrm{dec}'(\mathsf{sk}, \mathrm{enc}'(\mathsf{pk}, K)) \neq K$. Upon ERO, we immediately abort. It is noteworthy that GAME G_6 and GAME G_7 exhibit identical distributions when ERO does not occur (as implied by δ-correctness). Thus we have

$$\Pr[\mathrm{QUERY} : G_6] \leq \Pr[\mathrm{QUERY} : G_7] + \delta.$$

GAME G_8. In game G_8, the challenger simulates the $\mathsf{O}^{\mathrm{Dec}}(\mathrm{ct}_1, n_1)$ oracle without the secret key. Initially, we randomly choose $guess \leftarrow\!\!\$\ \{0,1\}$ to guess whether $K' = \mathrm{decaps}(\mathsf{sk}, \mathrm{ct}_1)$ equals to $\perp$. If $guess = 0$, we assume $K' = \perp$, and thus return $\perp$ in $\mathsf{O}^{\mathrm{Dec}}$. If $guess = 1$ and the corresponding $(\cdot, \mathrm{ct}_1, \cdot) \in \mathfrak{L}$, we directly extract the corresponding HS' from $\mathfrak{L}$. If there's no such $(\cdot, \mathrm{ct}_1, \cdot)$ in $\mathfrak{L}$, we can conclude that $\mathcal{A}$ has not queried $G(K')$ before based on rigid property of the DPKE. Assuming the adversary has queried $G(K')$ before, this implies the existence of $(K', \mathrm{ct}'_1, \cdot) \in \mathfrak{L}$, where $\mathrm{enc}'(\mathsf{pk}, K') = \mathrm{ct}'_1$. According to the rigid property, we have $\mathrm{ct}_1 = \mathrm{ct}'_1$. This contradicts the condition. Therefore, We just sample a uniformly random value $\mathrm{HS}' \leftarrow\!\!\$\ \{0,1\}^n$, and define $\mathfrak{L}^{\mathrm{Dec}} = (\mathrm{ct}_1, \mathrm{HS}')$. Finally, we utilize HS' to compute (tk_C, tk_S), thereby perfectly simulating $\mathsf{O}^{\mathrm{Dec}}$. To maintain consistency with $\mathsf{O}^{\mathrm{Dec}}$ and random oracle G, we change G as follow. When simulating $G(K)$ later, first compute $\mathrm{ct} = \mathrm{enc}'(\mathsf{pk}, K)$. If $\mathrm{ct} = \mathrm{ct}_1$, directly return HS' from $\mathfrak{L}_{\mathrm{Dec}}$ in this case. If $\mathrm{ct} \neq \mathrm{ct}_1$, we can assert that $\mathrm{dec}'(\mathsf{sk}, \mathrm{ct}) \neq \mathrm{dec}'(\mathsf{sk}, \mathrm{ct}_1)$ based on the rigid property.

Moreover, if the $(\mathrm{ct}_2, n_2, ., .)$ input to $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ is equal to the (ct_1, n_1) input to $\mathsf{O}^{\mathrm{Dec}}$, that is, $(\mathrm{ct}_1, n_1) = (\mathrm{ct}_2, n_2)$. If $guess = 0$, we directly return $\perp$ in $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$. Else we check if $(\cdot, \mathrm{ct}_1, \cdot) \in \mathfrak{L}$. If so, we find the corresponding HS'. Otherwise, we extract HS' from $\mathfrak{L}^{\mathrm{Dec}}$. Then we can use HS' to simulate $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$. Note that if ERO does not happen, this simulation is perfect when guessing correctly, thus we have

$$\Pr[\mathrm{QUERY} : G_7] = 2\Pr[\mathrm{QUERY} : G_8].$$

GAME G_9. In game G_9, the challenger can simulate $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}(\mathrm{ct}_2, \mathrm{n}_2, \mathrm{tag}, \mathrm{txt})$ without the secret key. We modify $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ as follows: If the $(\mathrm{ct}_2, n_2, ., .)$ input to $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ is not equal to the (ct_1, n_1) input to $\mathsf{O}^{\mathrm{Dec}}$ or $\mathcal{A}$ did not query $\mathsf{O}^{\mathrm{Dec}}$ before, we initially make another guess $guess' \leftarrow\!\!\$\ \{0,1\}$ to determine whether $K' = \perp$. If $guess' = 0$, we simply return $\perp$ in $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$. If $guess' = 1$ and the corresponding $(\cdot, \mathrm{ct}_2, \cdot) \in \mathfrak{L}$, we extract the corresponding HS'. Otherwise, we sample $\mathrm{HS}' \leftarrow\!\!\$\ \{0,1\}^n$ and define $\mathfrak{L}^{\mathrm{Dec}}_{\mathrm{MAC}} = (\mathrm{ct}_2, \mathrm{HS}')$. We then utilize HS' to perfectly simulate $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$. To maintain consistency with $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ and the random oracle G, when simulating $G(K)$ later, first compute $\mathrm{ct} = \mathrm{enc}'(\mathsf{pk}, K)$. If $\mathrm{ct} = \mathrm{ct}_2$, directly return HS' from $\mathfrak{L}^{\mathrm{MAC}}_{\mathrm{Dec}}$, otherwise, we simulate G as before. Note that this analysis is identical to G_8. It is apparent that based on the rigid property of DPKE this simulation is perfect when guessing correctly. Thus, we have

$$\Pr[\mathrm{QUERY} : G_8] = 2\Pr[\mathrm{QUERY} : G_9].$$

Now, we construct an OW-CPA adversary $\mathcal{C}(\mathsf{pk}, \mathrm{ct}^*)$ against the DKEM. $\mathcal{C}$ samples $guess, guess', n^*$ as in G_9 and $\mathrm{CHTS}^*, \mathrm{SHTS}^*, \mathrm{dHS}^* \leftarrow\!\!\$\ \{0,1\}^{3n}$. Then $\mathcal{C}$ picks six $2q_{H_k}$-wise ($k \in \{1,2,3,4,H,T\}$) independent functions and runs $\mathcal{A}^{G,H_i,\mathsf{O}^{\mathrm{Dec}},\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}}(pk, \mathrm{ct}^*, n^*, \mathrm{CHTS}^*, \mathrm{SHTS}^*, \mathrm{dHS}^*)$ as in game G_9, lastly selects a K' from G-List such that $\mathrm{enc}'(K') = \mathrm{ct}^*$, and returns K'. Note that if ERO does not happen, $\mathcal{C}$ returns K^* with probability $\Pr[\mathrm{QUERY} : G_9]$. Thus $\mathrm{Adv}^{\text{OW-CPA}}_{\mathrm{DKEM}}(\mathcal{C}) \geq \Pr[\mathrm{QUERY} : G_9]$. Putting everything together, we have

$$\begin{aligned}\mathrm{Adv}^{\text{IND-1CCA-MAC}}_{\mathrm{KEM}}(\mathcal{A}) \leq & 4\mathrm{Adv}^{\text{OW-CPA}}_{\mathrm{DKEM}}(\mathcal{C}) + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}) \\ & + \frac{q_{H_1} + 2q_{H_2} + q_{H_3} + q_{H_D} + q_{H_T} + 6}{2^n} + \delta\end{aligned}$$

□

Theorem 4. *Let* $\mathrm{KEM} = (\mathrm{gen}, \mathrm{encaps}, \mathrm{decaps}) = T(\mathrm{DPKE}(\mathrm{gen}', \mathrm{enc}', \mathrm{dec}'))$ *be a DKEM and the underlying* δ*-correctness DPKE is rigid. Let the KDFs and the hash function in the* IND-1CCA-MAC* *game be modeled as random oracles. Then, for any PPT adversary* $\mathcal{A}$ *making at most* q_G, q_{H_1}, q_{H_2}, q_{H_3}, q_{H_4}, q_{H_D}, q_{H_T} *queries to* G, H_1, H_2, H_3, H_4, H_D, H_T *respectively, there exists an OW-CPA adversary* $\mathcal{C}$ *and an EUF-0T adversary* $\mathcal{B}$ *such that*

$$\begin{aligned}\mathrm{Adv}^{\text{IND-1CCA-MAC*}}_{\mathrm{KEM}}(\mathcal{A}) \leq & 2\mathrm{Adv}^{\text{OW-CPA}}_{\mathrm{DKEM}}(\mathcal{C}) + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}) \\ & + \frac{q_{H_1} + 2q_{H_2} + q_{H_3} + q_{H_D} + q_{H_T} + 6}{2^n} + \delta\end{aligned}$$

where $\mathcal{C}$ *have approximately the same running time as* $\mathcal{A}$.

Proof. It is easy to see that except for G_9, the proof is the same as the proof of the theorem 3. In the proof of IND-1CCA-MAC*, we can simply define G_9 to be identical to G_8. Thus, in this proof we have $\Pr[\mathrm{QUERY} : G_8] = \Pr[\mathrm{QUERY} : G_9]$. □

3.2 OW-CPA/IND-CPA/D-OW-CPA KEMs imply IND-1CCA-MAC* in the QROM

We now prove that any OW-CPA/IND-CPA/D-OW-CPA KEMs are also IND-1CCA-MAC* secure in the QROM with an EUF-0T secure MAC.

Theorem 5. *Let* $\mathrm{KEM} = (\mathrm{gen}, \mathrm{encaps}, \mathrm{decaps})$ *be a KEM. Let the KDFs and the hash function in the* IND-1CCA-MAC* *game be modeled as quantum random oracles. Then, for any PPT adversary* $\mathcal{A}$ *issuing at most one single (classical) query to the* $\mathsf{O}^{\mathrm{Dec}}$, $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ *oracle and making at most* q_G, q_{H_1}, q_{H_2}, q_{H_3}, q_{H_4}, q_{H_D}, q_{H_T} *queries to* G, H_1, H_2, H_3, H_4, H_D, H_T *respectively and let* $q_{123} = q_{H_1} + q_{H_2} + q_{H_3} + 1$, *there exists a PPT OW-CPA adversary* $\mathcal{C}$, *a PPT D-OW-CPA adversary* $\hat{\mathcal{C}}$ *(if KEM is DPKE), a PPT IND-CPA adversary* $\mathcal{D}$, *a PPT EUF-0T adversary* $\mathcal{B}_1$ *and a PPT EUF-0T adversary* $\mathcal{B}_2$ *such that*

$$\begin{aligned}
&\mathrm{Adv}^{\text{IND-1CCA-MAC}^*}_{\mathrm{KEM}}(\mathcal{A}) \\
&\leq \frac{2q_{123}+6q_{H_4}(q_{H_D}+2)}{2^{n/2}} + \frac{(q_{H_T}+4)^2+(q_{H_2}+1)^2}{2^n} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_1) \\
&\quad + 8q_G(q_{H_1}+q_{H_2}+1)\cdot \\
&\qquad \sqrt{\mathrm{Adv}^{\text{OW-CPA}}_{\mathrm{KEM}}(\mathcal{C}) + \frac{1}{2^{2n}} + 6q_{H_4}(q_{H_D}+2)2^{-n/2} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_2)}.
\end{aligned}$$

$$\begin{aligned}
&\mathrm{Adv}^{\text{IND-1CCA-MAC}^*}_{\mathrm{KEM}}(\mathcal{A}) \\
&\leq \frac{2q_{123}+6q_{H_4}(q_{H_D}+2)}{2^{n/2}} + \frac{(q_{H_T}+4)^2+(q_{H_2}+1)^2}{2^n} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_1) \\
&\quad + 8(q_{H_1}+q_{H_2}+1)\cdot \\
&\qquad \sqrt{\mathrm{Adv}^{\text{OW-CPA}}_{\mathrm{DKEM}}(\hat{\mathcal{C}}) + \frac{1}{2^{2n}} + \delta + 6q_{H_4}(q_{H_D}+2)2^{-n/2} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_2)}.
\end{aligned}$$

$$\begin{aligned}
&\mathrm{Adv}^{\text{IND-1CCA-MAC}^*}_{\mathrm{KEM}}(\mathcal{A}) \\
&\leq \frac{2q_{123}+6q_{H_4}(q_{H_D}+2)}{2^{n/2}} + \frac{(q_{H_T}+4)^2+(q_{H_2}+1)^2}{2^n} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_1) \\
&\quad + 8(q_{H_1}+q_{H_2}+1)\cdot \\
&\qquad \sqrt{2\mathrm{Adv}^{\text{IND-CPA}}_{\mathrm{KEM}}(\mathcal{D}) + \frac{(q_G+1)^2}{|\mathcal{K}|} + 6q_{H_4}(q_{H_D}+3)2^{-n/2} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_2)}.
\end{aligned}$$

where $\mathcal{B}_1$, $\mathcal{B}_2$, $\mathcal{C}$, and $\mathcal{D}$ have approximately the same running time as $\mathcal{A}$.

Proof sketch: The proof consists of four main steps. The first is embedding the underlying hard problem by replacing the real key HS^* with a random value. When the underlying KEM is OW-CPA-secure, we use general OW2H (Lemma 3) to argue the reprogramming impact. When the KEM is IND-CPA-secure or D-OW-CPA-secure, we use the double-sided OW2H (Lemma 4) to discuss the impact of reprogramming. Since the embedded IND-CPA game is decisional, we also need to use Lemma 5 to argue the advantage that searching for a reprogramming point results in a double-sided oracle.

The second step is simulate the $\mathsf{O}^{\mathrm{Dec}}(\overline{\mathrm{ct}}, \overline{n})$ oracle without the secret key. Initially, we utilize an internal hash function $H_{12} = (H_1, H_2)$ to simulate the quantum random oracles H_1 and H_2. We first randomly choose $guess \leftarrow\!\$\ \{0,1\}$ to guess whether $\mathrm{decaps}(\mathsf{sk}, \overline{\mathrm{ct}}) = \perp$. When $guess = 0$, we return $\perp$. When $guess = 1$, we proceed with the simulation as follows. As discussed in Sec. 1.3, we directly reprogram $H_{12}(\overline{\mathrm{HS}}, \bar{t})$ to random values $\Theta = (\overline{chts}, \overline{shts})$, where $\overline{\mathrm{HS}} = G(K)$ and $\bar{t} = H_T(\overline{\mathrm{ct}}, \overline{n})$, then output $H_D(\overline{chts}), H_D(\overline{shts})$ in the $\mathsf{O}^{\mathrm{Dec}}$. Intuitionally, the simulation is perfect if we reprogram $H_{12}(\overline{\mathrm{HS}}, \bar{t})$ to Θ when the adversary $\mathcal{A}$ first queries $(\overline{\mathrm{HS}}, \bar{t})$ to either H_1 or H_2 in the ROM. However,

in the QROM, it is hard to define when the adversary makes a query $(\overline{\mathrm{HS}}, \bar{t})$ to H_1 or H_2. Therefore, in the QROM, we use the idea from [24] to argue it differently. As discussed in Sec. 1.3 we find that the simulation is perfect if the predicate $\mathsf{O}^{\mathrm{Dec}}(\overline{\mathrm{ct}}, \overline{n}) = (H_D(H_1(\overline{\mathrm{HS}}, \bar{t})), H_D(H_2(\overline{\mathrm{HS}}, \bar{t})))$ is satisfied. Since in the practical implementation of the $\mathsf{O}^{\mathrm{Dec}}(\overline{\mathrm{ct}}, \overline{n})$ oracle, there is an implicit classical query to H_{12}, which is removed during the oracle's simulation in $\mathsf{O}^{\mathrm{Dec}}$. Therefore, this specific query cannot be measured. Hence, we employ the refined optional-query measure-and-reprogram technique [24] (Lemma 6) to argue the simulation impact.

The third step is simulate the $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}(\overline{\mathrm{ct}}, \overline{n}, \mathrm{txt}, \mathrm{tag})$ oracle without the secret key. Recall that $(\overline{\mathrm{ct}}, \overline{n})$ remains consistent with the $\mathsf{O}^{\mathrm{Dec}}$ oracle according to the definition in IND-1CCA-MAC*. When the *guess* value in $\mathsf{O}^{\mathrm{Dec}}$ is 0, we return $\perp$ in $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$. Otherwise, we proceed with the following simulation. While simulating $\mathsf{O}^{\mathrm{Dec}}(\overline{\mathrm{ct}}, \overline{n})$, we employ the refined optional-query measure and reprogram technique on H_{12}, where the measurement yields x, and then reprogram H_{12} on x to Θ. If MAC.Vrf(fk_S, txt, tag) is correct, we need to return HS in $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$, where $fk_S = H_D(\overline{shts})$. Intuitively, we can directly use x (which includes HS) to output HS. However, it is crucial to discuss the order of $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ query and the measurement in the refined optional-query measure and reprogram technique. If $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ query occurs after the measurement, we can utilize x to perfectly simulate it by return HS. However, if $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ query occurs before the measurement, we cannot return HS since we have not yet measured it to obtain x. Nonetheless, this scenario implies that all H_{12} queries made by the adversary so far are not reprogrammed and unrelated to the reprogrammed values Θ, which are utilized to derive the MAC key. Thus, we can argue that the adversary $\mathcal{A}$ cannot distinguish fk_S from a random value using OW2H Lemma. Hence, according to the security of MAC, the adversary cannot forge a valid tag. Therefore, if $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ query occurs before the measurement, we directly output $\perp$.

Finally, we reprogram $(\mathrm{CHTS}^*_0, \mathrm{SHTS}^*_0, \mathrm{dHS}^*_0) = (H_1(\mathrm{HS}^*, t^*), H_2(\mathrm{HS}^*, t^*), H_3(\mathrm{HS}^*))$ to random values. In this case, the adversary's probability of winning this game is $1/2$. We then utilize HS^* to be indistinguishable from random for the adversary to analyze the impact of such reprogramming using OW2H Lemma.

Proof. We provide the complete proof in Appendix A. □

Remark 2. One can demonstrate the security of the IND-1CCA-MAC based on CPA-secure KEMs in the QROM using a similar proof technique as Theorem 5 with greater loss reduction compared to IND-1CCA-MAC* because it is necessary to utilize the Measure-and-Reprogram technique twice (Lemma 6).

3.3 Multi-Stage Security for TLS 1.3 from IND-1CCA-MAC*

We now demonstrate that if the IND-1CCA-MAC* is satisfied, the 1-RTT TLS 1.3 handshake is secure in the MultiStage model. Theorem 6 aligns with Theorem 5 in [22], but substitutes the IND-1CCA-MAC with the IND-1CCA-MAC*.

Theorem 6. *The TLS 1.3 full 1-RTT handshake is secure in the MultiStage model if the underlying KEM is* IND-1CCA-MAC* *(and the signature is secure). Formally for any Multi-Stage PPT adversary $\mathcal{A}$, there exist PPT adversaries $\{\mathcal{B}_i\}_{i\in[6]}$ such that*

$$\mathrm{Adv}^{\text{multi-stage}}_{\text{TLS1.3-1RTT}}(\mathcal{A}) \leq 6t_s \left(\begin{array}{l} \mathrm{Adv}^{\text{coll}}_{\text{H}}(\mathcal{B}_1) + t_u \mathrm{Adv}^{\text{euf-cma}}_{\text{Sig}}(\mathcal{B}_2) \\ + t_s \left(\begin{array}{l} \mathrm{Adv}^{\text{ind-1cca-mac}^*}_{\text{KEM}}(\mathcal{B}_3) + 2 \cdot \mathrm{Adv}^{\text{prf}}_{\text{HKDF.Exp}}(\mathcal{B}_4) \\ + \mathrm{Adv}^{\text{prf}}_{\text{HKDF.Ext}}(\mathcal{B}_5) + \mathrm{Adv}^{\text{prf}}_{\text{HKDF.Exp}}(\mathcal{B}_6) \end{array} \right) \end{array} \right)$$

where t_s (resp. t_u) is the maximal number of sessions (resp. users).

Proof Sketch: This proof is identical to the proof in [22], Theorem 5, except for replacing IND-1CCA-MAC with IND-1CCA-MAC*. For detailed steps, please refer to the original proof in [17], Theorem 6.4. They utilized the snPRF-ODH assumption to substitute HS with a random $\widetilde{\text{HS}}$ [17]. Following [22], we directly apply the IND-1CCA-MAC* to concurrently replace CHTS, SHTS, and dHS with random values, and subsequently leverages PRF properties to replace additional keys with random values. In particular, IND-1CCA-MAC* is sufficient for the security of TLS 1.3. The complete proof is provided in Appendix C.

Theorem 7. *The modified TLS 1.3 handshake in the pre-shared key (optional) 0-RTT mode with key exchange (i.e., TLS 1.3 PSK-(EC)-DHE 0-RTT) is secure in the MultiStage model if the underlying KEM is* IND-1CCA-MAC* *(and signature, MAC, etc. are secure), in the sense of Dowling et al. [17].*

We provide the complete statement and a proof sketch for Theorem 7 in Appendix D.

Remark 3. Combining Theorem 2, 4, 5, 6, 7, we obtain the security proof of TLS 1.3 from OW-CPA/IND-CPA/D-OW-CPA KEMs (with a secure MAC) in the ROM and QROM.

Acknowledgements. We would like to thank anonymous reviewers of Asiacrypt 2024 for their insightful comments and suggestions. We thank Jieyu Zheng for her help in benchmarking. Biming Zhou and Yunlei Zhao was supported by the National Key R&D Program of China (No. 2022YFB2701601), General Project of State Key Laboratory of Cryptography (No. MMKFKT202227), Technical Standard Project of Shanghai Scientific and Technological Committee (No. 21DZ2200500), Shanghai Collaborative Innovation Fund (No. XTCX-KJ-2023-54), and Special Fund for Key Technologies in Blockchain of Shanghai Scientific and Technological Committee (No. 23511100300). Haodong Jiang was supported by the National Key R&D Program of China (No. 2021YFB3100100), and the National Natural Science Foundation of China (No. 62002385)

A Supporting Material: Proof of Theorem 5

A.1 Quantum Random Oracle Model

We refer the reader to [32] for the basics of quantum computation and quantum information. The Random Oracle Model (ROM) [6] is an ideal model where a uniformly random function is selected and publicly accessible. In the quantum setting, a quantum adversary can evaluate the hash function on arbitrary superposition inputs. Therefore, in the Quantum Random Oracle Model (QROM), we model that a quantum adversary is allowed to query the random oracle with quantum states [9]. We introduce several Lemmas that are utilized throughout this proof.

Lemma 1. *(Simulating the random oracle [38], Theorem 6.1). Let H be an oracle drawn from the set of $2q$-wise independent functions uniformly at random. Then, the advantage any quantum algorithm making at most q queries to H has in distinguishing H from a truly random function is identically 0.*

Lemma 2. *(Generic search problem [23], Lemma 3). Let $\gamma \in [0,1]$. Let Z be a finite set. $N_1 : Z \to \{0,1\}$ is the following function: For each z, $N_1(z) = 1$ with probability p_z ($p_z \leq \gamma$), and $N_1(z) = 0$ else. Let N_2 be the function with $\forall z : N_2(z) = 0$. If an oracle algorithm A makes at most q quantum queries to N_1 (or N_2), then*

$$\left|\Pr[b = 1 : b \leftarrow A^{N_1}] - \Pr[b = 1 : b \leftarrow A^{N_2}]\right| \leq 8(q+1)^2\gamma.$$

Particularly, the probability of A finding a z such that $N_1(z) = 1$ is at most $8(q+1)^2\gamma$, i.e., $\Pr[N_1(z) = 1 : z \leftarrow A^{N_1}] \leq 8(q+1)^2\gamma$.

Lemma 3. *(One-way to hiding (OW2H) [3], Theorem 3). Let $S \subseteq X$ be random. Let G, H be oracles such that $\forall x \notin S.\ G(x) = H(x)$. Let z be a random bitstring. (S, G, H, z may have arbitrary joint distribution.) Let A be a quantum oracle algorithm that makes at most q queries (not necessarily unitary). Let $B^{|H\rangle}$ be an oracle algorithm that, on input z, does the following: pick $i \in [q-1]$, run $A^{|H\rangle}(z)$ until (just before) the $(i+1)$-th query, measure all query input registers in the computational basis, and output the set T of measurement outcomes. Then*

$$\left|\Pr[1 \leftarrow A^{|H\rangle}(z)] - \Pr[1 \leftarrow A^{|G\rangle}(z)]\right| \leq 2q\sqrt{\Pr[S \cap T \neq \varnothing : T \leftarrow B^{|H\rangle}(z)]}.$$

Lemma 4. *((Adapted) Double-sided O2H [8], Lemma 5). Let G, $H : \mathcal{X} \to \mathcal{Y}$ be oracles such that $\forall x \neq x^* :\ G(x) = H(x)$. Let z be a random bitstring. (x^*, G, H, z may have arbitrary joint distribution.) Let A be a quantum oracle algorithm that makes at most q queries (not necessarily unitary). Then, there is another double-sided oracle algorithm $B^{|G\rangle,|H\rangle}(z)$ such that B runs in about the same amount of time as A, and*

$$\left|\Pr[1 \leftarrow A^{|H\rangle}(z)] - \Pr[1 \leftarrow A^{|G\rangle}(z)]\right| \leq 2\sqrt{\Pr[x^* = x' : x' \leftarrow B^{|G\rangle,|H\rangle}(z)]}.$$

In particular, the double-sided oracle algorithm $B^{|G\rangle,|H\rangle}(z)$ runs $A^{|H\rangle}(z)$ and $A^{|G\rangle}(z)$ in superposition, and the probability $\Pr[x^* = x' : x' \leftarrow B^{|G\rangle,|H\rangle}(z)]$ is exactly $\||\psi_q^H\rangle - |\psi_q^G\rangle\|^2/4$, where $|\psi_q^H\rangle$ ($|\psi_q^G\rangle$, resp.) is the final state of $A^{|H\rangle}(z)$ ($A^{|G\rangle}(z)$, resp.).

Lemma 5. *(Search in Double-sided Oracle [24], Lemma 2.3) . Let $G, H : \mathcal{X} \to \mathcal{Y}$ be oracles such that $\forall x \neq x^* : \ G(x) = H(x)$. Let z be a random bitstring. Let A be a quantum oracle algorithm that makes at most q queries (not necessarily unitary). Let $B^{|G\rangle,|H\rangle}(z)$ be a double-sided oracle algorithm such that $\Pr[x^* = x' : x' \leftarrow B^{|G\rangle,|H\rangle}(z)] = \||\psi_q^H\rangle - |\psi_q^G\rangle\|^2/4$, where $|\psi_q^H\rangle$ ($|\psi_q^G\rangle$, resp.) is the final state of $A^{|H\rangle}(z)$ ($A^{|G\rangle}(z)$, resp.). Let $C^{|H\rangle}(z)$ be an oracle algorithm that picks $i \leftarrow \{1, 2, \ldots, q\}$, runs $A^{|H\rangle}(z)$ until (just before) the i-th query, measures the query input registers in the computational basis, and outputs the measurement outcome. Thus, we have*

$$\Pr[x^* = x' : x' \leftarrow B^{|G\rangle,|H\rangle}(z)] \leq q^2 \Pr[x^* = x' : x' \leftarrow C^{|H\rangle}(z)].$$

In particular, if $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$, $x^* = (x_1^*, x_2^*)$, x_1^* is uniform and independent of H and z, then we further have

$$\Pr[x^* = x' : x' \leftarrow B^{|G\rangle,|H\rangle}(z)] \leq \frac{q^2}{|\mathcal{X}_1|}.$$

Measure-and-Reprogram, introduced by [14, 15], demonstrates how to adaptively reprogram the quantum random oracle at a single input. Specifically, for any oracle algorithm $A^{|H\rangle}$ that makes at most q queries to H and outputs a pair (x, z) such that some predicate $V(x, H(x), z)$ holds true, the Measure-and-Reprogram technique demonstrates the existence of another algorithm S^A that emulates H, extracts x from $A^{|H\rangle}$ by randomly measuring one of A's queries to H, and subsequently reprograms $H(x)$ to a designated value Θ, ensuring that the output z from $A^{|H\rangle}$ satisfies $V(x, \Theta, z)$ with a multiplicative $O(q^2)$ loss in probability. Due to the presence of an implicit classical H-query in the proof (which corresponds exactly to x), we need to employ a variant of the Measure-and-Reprogram technique proposed by [24] in the proof. Informally, this variant of the Measure-and-Reprogram technique states that if for any algorithm $A^{|H\rangle}$ some predicate V holds true, one can build another algorithm S^A that does not query H on the i^*-th query (but uses its input instead) can satisfy V with a multiplicative $O(q^2)$ loss in probability.

Lemma 6. *(Single-classical-query Measure-and-Reprogram [24], Lemma 3.1). Let $A^{|H\rangle}$ be an arbitrary oracle quantum algorithm that makes q queries to a uniformly random function $H : X \to Y$ and outputs some classical $x \in X$ and a (possibly quantum) output z. In particular, the i^*-th query input state of A is $|x\rangle$ (this is a classical state and identical with the x output by $A^{|H\rangle}$).*

Let $S^A(\Theta)$ be an oracle algorithm that randomly picks a pair $(i, b_0) \in ([q-1] \setminus \{i^*-1\} \times \{0,1\}) \cup \{(q, 0)\}$, runs $A^{|H_i^{i^*}\rangle}$ to output z, where $H_i^{i^*}$ is an oracle that returns Θ for A's i^*-th H query, measures A's $(i+1)$-th query input to

obtain x, returns A's l-th query to H for $l < (i+1+b_0)$ and $l \neq i^*$, and returns A's l-th query to $H_{x\Theta}$ ($H_{x\Theta}(x) = \Theta$ and $H_{x\Theta}(x') = H(x')$ for all $x' \neq x$) for $l \geq (i+1+b_0)$ and $l \neq i^*$.

Let $S_1^A(\Theta)$ be an oracle algorithm that randomly picks a pair $(j, b_1) \in (\{i^*, \ldots, q-1\} \times \{0,1\}) \cup \{(q,0)\} \cup \{(i^*-1, 1)\}$, runs $A^{|H_j\rangle}$ to output z, where H_j is an oracle that measures A's $(j+1)$-th query input to obtain x, returns A's l-th query to H for $l < (j+1+b_1)$, and returns A's l-th query to $H_{x\Theta}$ for $l \geq (j+1+b_1)$.

Thus, for any $x_0 \in X$, $i^* \in \{1, \ldots, q\}$, and any predicate V :

$$\begin{aligned}
&\Pr_H[x = x_0 \wedge V(x, H(x), z) = 1 : (x, z) \leftarrow A^{|H\rangle}] \\
&\leq 2(2q-1)^2 \Pr_{H,\Theta}[x = x_0 \wedge V(x, \Theta, z) = 1 : (x, z) \leftarrow S^A] \\
&\quad + 8q^2 \Pr_{H,\Theta}[x = x_0 \wedge V(x, \Theta, z) = 1 : (x, z) \leftarrow S_1^A],
\end{aligned}$$

where the subscript $\{H, \Theta\}$ in $\Pr_H$ and $\Pr_{H,\Theta}$ denotes that the probability is averaged over a random choice of H and Θ. Moreover, if $V = V_1 \wedge V_2$ such that $V_1(x, y, z) = 1$ iff y is returned for A's i^*-th query, then $\sum x_0 \Pr_{H,\Theta}[x = x_0 \wedge V(x, \Theta, z) = 1 : (x, z) \leftarrow S_1^A] \leq \frac{1}{|\mathcal{Y}|}$.

A.2 Proof

Theorem 5. *Let* $\mathrm{KEM} = (\mathrm{gen}, \mathrm{encaps}, \mathrm{decaps})$ *be a KEM. Let the KDFs and the hash function in the* IND-1CCA-MAC* *game be modeled as quantum random oracles. Then, for any PPT adversary $\mathcal{A}$ issuing at most one single (classical) query to the $\mathsf{O}^{\mathrm{Dec}}$, $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ oracle and making at most q_G, q_{H_1}, q_{H_2}, q_{H_3}, q_{H_4}, q_{H_D}, q_{H_T} queries to G, H_1, H_2, H_3, H_4, H_D, H_T respectively and let $q_{123} = q_{H_1} + q_{H_2} + q_{H_3} + 1$, there exists a PPT OW-CPA adversary $\mathcal{C}$, a PPT D-OW-CPA adversary $\hat{\mathcal{C}}$ (if KEM is DPKE), a PPT IND-CPA adversary $\mathcal{D}$, a PPT EUF-0T adversary $\mathcal{B}_1$ and a PPT EUF-0T adversary $\mathcal{B}_2$ such that*

$$\begin{aligned}
&\mathrm{Adv}^{\text{IND-1CCA-MAC}^*}_{\mathrm{KEM}}(\mathcal{A}) \\
&\leq \frac{2q_{123} + 6q_{H_4}(q_{H_D}+2)}{2^{n/2}} + \frac{(q_{H_T}+4)^2 + (q_{H_2}+1)^2}{2^n} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_1) \\
&\quad + 8q_G(q_{H_1} + q_{H_2} + 1) \cdot \\
&\qquad \sqrt{\mathrm{Adv}^{\text{OW-CPA}}_{\mathrm{KEM}}(\mathcal{C}) + \frac{1}{2^{2n}} + 6q_{H_4}(q_{H_D}+2)2^{-n/2} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_2)}.
\end{aligned}$$

$$
\begin{aligned}
&\mathrm{Adv}^{\text{IND-1CCA-MAC}^*}_{\mathrm{KEM}}(\mathcal{A}) \\
&\leq \frac{2q_{123} + 6q_{H_4}(q_{H_D}+2)}{2^{n/2}} + \frac{(q_{H_T}+4)^2 + (q_{H_2}+1)^2}{2^n} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_1) \\
&\quad + 8(q_{H_1}+q_{H_2}+1)\cdot \\
&\qquad \sqrt{\mathrm{Adv}^{\text{OW-CPA}}_{\mathrm{DKEM}}(\hat{\mathcal{C}}) + \frac{1}{2^{2n}} + \delta + 6q_{H_4}(q_{H_D}+2)2^{-n/2} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_2)}.
\end{aligned}
$$

$$
\begin{aligned}
&\mathrm{Adv}^{\text{IND-1CCA-MAC}^*}_{\mathrm{KEM}}(\mathcal{A}) \\
&\leq \frac{2q_{123} + 6q_{H_4}(q_{H_D}+2)}{2^{n/2}} + \frac{(q_{H_T}+4)^2 + (q_{H_2}+1)^2}{2^n} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_1) \\
&\quad + 8(q_{H_1}+q_{H_2}+1)\cdot \\
&\qquad \sqrt{2\mathrm{Adv}^{\text{IND-CPA}}_{\mathrm{KEM}}(\mathcal{D}) + \frac{(q_G+1)^2}{|\mathcal{K}|} + 6q_{H_4}(q_{H_D}+3)2^{-n/2} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_2)}.
\end{aligned}
$$

where $\mathcal{B}_1$, $\mathcal{B}_2$, $\mathcal{C}$, and $\mathcal{D}$ have approximately the same running time as $\mathcal{A}$.

Proof. Let $\mathcal{A}$ be a quantum adversary against the IND-1CCA-MAC* game, issuing (exactly) one classical query to $\mathsf{O}^{\mathrm{Dec}}(\overline{\mathrm{ct}}, \overline{n})$ and then one classical query to $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}(\overline{\mathrm{ct}}, \overline{n}, \mathrm{tag}, \mathrm{txt})$ which is consistent with $\mathsf{O}^{\mathrm{Dec}}$ (by introducing a dummy query if necessary). Consider the games in Fig. 10,11.

GAME G_0: This is the original IND-1CCA-MAC* game. From now on, unless otherwise specified, all hash queries are quantum queries. Therefore, $|\Pr[G_0^A \Rightarrow 1] - 1/2| = \mathrm{Adv}^{\mathrm{KEM}}_{\text{IND-1CCA-MAC}^*}(\mathcal{A})$.

GAME G_1: In game G_1, we modify the previous game as follows. We abort if, in the $\mathsf{O}^{\mathrm{Dec}}(\overline{\mathrm{ct}}, \overline{n})$ oracle, the calculation $H_T(\overline{\mathrm{ct}}, \overline{n}) = H_T(\mathrm{ct}^*, n^*)$ holds, but $(\overline{\mathrm{ct}}, \overline{n}) \neq (\mathrm{ct}^*, n^*)$. Since there are at most $q_{H_T}+4$ queries to H_T in the game, the probability that $\mathcal{A}$ finds such $(\overline{\mathrm{ct}}, \overline{n})$ given by (ct^*, n^*) is less than $\epsilon = \frac{(q_{H_T}+4)^2}{2^n}$ ([39], Corollary 1). Consequently, we have

$$
|\Pr[G_0^{\mathcal{A}} \Rightarrow 1] - \Pr[G_1^{\mathcal{A}} \Rightarrow 1]| \leq \frac{(q_{H_T}+4)^2}{2^n}
$$

GAME G_2: In game G_2, We abort if the value SHTS computed in the $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ oracle is such that $\mathrm{SHTS} = \mathrm{SHTS}_b$ (classical). Since G_1 and G_2 return $\perp$ when $H_T(\overline{\mathrm{ct}}, \overline{n}) = H_T(\mathrm{ct}^*, n^*)$, the event $\mathrm{SHTS} = \mathrm{SHTS}_b = H_2(., H_T(\mathrm{ct}^*, n^*))$ occurs implies the adversary that given SHTS_b finds $H_2(., H_T(\overline{\mathrm{ct}}, \overline{n})) = \mathrm{SHTS}_b$ such that $H_T(\overline{\mathrm{ct}}, \overline{n}) \neq H_T(\mathrm{ct}^*, n^*)$. Note that there are at most $q_{H_2}+1$ queries to H_2 in the game, the probability of athe event $\mathrm{SHTS} = \mathrm{SHTS}_b$ occurring is at most $\frac{(q_{H_2}+1)^2}{2^n}$. Hence, we have

$$
|\Pr[G_1^{\mathcal{A}} \Rightarrow 1] - \Pr[G_2^{\mathcal{A}} \Rightarrow 1]| \leq \frac{(q_{H_2}+1)^2}{2^n}.
$$

GAMES $G_0 - G_9$

$G, H_{i\in\{1-4,T,D\}} \leftarrow\$\ \Omega_G, \Omega_{H_i}, H_{12} \leftarrow\$\ \Omega_{H_{12}}, H_{123} \leftarrow\$\ \Omega_{H_{123}}$

$b \leftarrow\$\ \{0,1\}, (\mathsf{pk},\mathsf{sk}) \leftarrow \text{gen}, (\text{ct}^*, K^*) \leftarrow\$\ \text{encaps}(pk)$

$n^*, \text{HS}_1, \text{CHTS}_1, \text{SHTS}_1, \text{dHS}_1 \leftarrow\$\ \{0,1\}^n$

$\text{HS}^* \leftarrow G(K^*)$ $//G_0 - G_4$

$\text{HS}^* \leftarrow\$\ \{0,1\}^n$ $//G_5-$

use $H_{12} = (H_1, H_2)$ to simulate H_1, H_2 $//G_3 - G_6$

use $H_{123} = (H_1, H_2, H_3)$ to simulate H_1, H_2, H_3 $//G_7, G_9$

use H'_{123} to simulate H_1, H_2, H_3 $//G_8$

$\text{CHTS}_0 \leftarrow H_1(\text{HS}^*, H_T(\text{ct}^*, n^*)), \text{SHTS}_0 \leftarrow H_2(\text{HS}^*, H_T(\text{ct}^*, n^*))$

$\text{dHS}_0 \leftarrow H_3(\text{HS}^*)$

$(\text{CHTS}_0, \text{SHTS}_0, \text{dHS}_0) \leftarrow\$\ \{0,1\}^{3n}$ $//G_9$

$(\text{CHTS}_1, \text{SHTS}_1, \text{dHS}_1) \leftarrow\$\ \{0,1\}^{3n}$

$b' \leftarrow \mathcal{A}^{|G\rangle,|H_i\rangle,\mathsf{O}^{\text{Dec}},\mathsf{O}^{\text{Dec}}_{\text{MAC}}}(\mathsf{pk}, \text{ct}^*, n^*, (\text{CHTS}_b, \text{SHTS}_b, \text{dHS}_b))$ $//G_{0-3,5-9}$

$b' \leftarrow \mathcal{A}^{|G'\rangle,|H_i\rangle,\mathsf{O}^{\text{Dec}},\mathsf{O}^{\text{Dec}}_{\text{MAC}}}(\mathsf{pk}, \text{ct}^*, n^*, (\text{CHTS}_b, \text{SHTS}_b, \text{dHS}_b))$ $//G_4$

return $1_{b=b'}$

$\mathsf{O}^{\text{Dec}}(\overline{\text{ct}}, \overline{n})$

if more than 1 query : **return** $\perp$

if $(\overline{\text{ct}}, \overline{n}) = (\text{ct}^*, n^*)$: **return** $\perp$

if $H_T(\overline{\text{ct}}, \overline{n}) = H_T(\text{ct}^*, n^*)$: // G_1-

  abort

$K' \leftarrow \text{decaps}(\mathsf{sk}, \overline{\text{ct}})$

if $K' = \perp$: **return** $\perp$

$\text{HS}' \leftarrow G(K')$

if $K' = K^*$: $\text{HS}' = \text{HS}^*$ $//G_5-$

$\text{CHTS} \leftarrow H_1(\text{HS}', H_T(\overline{\text{ct}}, \overline{n}))$

$\text{SHTS} \leftarrow H_2(\text{HS}', H_T(\overline{\text{ct}}, \overline{n}))$

$\text{tk}_C \leftarrow H_D(\text{CHTS})$

$\text{tk}_S \leftarrow H_D(\text{SHTS})$

return tk_C, tk_S

$\mathsf{O}^{\text{Dec}}_{\text{MAC}}(\overline{\text{ct}}, \overline{n}, \text{tag}, \text{txt})$

if more than 1 query : **return** $\perp$

if $\mathcal{A}$ did not query O^{Dec} : **return** $\perp$

if $(\overline{\text{ct}}, \overline{n}) = (\text{ct}^*, n^*)$: **return** $\perp$

$K' \leftarrow \text{decaps}(\mathsf{sk}, \overline{\text{ct}})$

if $K' = K^*$: $\text{HS}' = \text{HS}^*$ $//G_5$

if $K' = K^*$: **return** $\perp$ $//G_6-$

$\text{HS}' \leftarrow G(K')$

$\text{SHTS} \leftarrow H_2(\text{HS}', H_T(\overline{\text{ct}}, \overline{n}))$

$\text{fk}_S \leftarrow H_4(\text{SHTS})$

if $\text{SHTS} = \text{SHTS}_b$: **abort** // G_2-

if $\text{MAC.Vrf}(\text{fk}_S, \text{txt}, \text{tag}) = \text{true}$:

  return HS'

return $\perp$

Fig. 10: Games $G_0 - G_9$ for the proof of Theorem 5

$G'(K)$ // G_4	$H'_{123}(\mathrm{HS}, t)$ // G_8
1 : **if** $K = K^*$: **return** HS_1	1 : **if** $(\mathrm{HS}, t) = (\mathrm{HS}^*, t^*)$:
2 : **return** $G(K)$	2 : **return** $(\mathrm{CHTS}_1, \mathrm{SHTS}_1, \mathrm{dHS}_1)$
	3 : **return** $H_{123}(\mathrm{HS}, t)$

Fig. 11: Games G_4, G_8 for the proof of Theorem 5

GAME G_3: In game G_3, we use a new quantum random oracle $H_{12}(\mathrm{HS}, t)$ to simulate both H_1 and H_2. Let $H_{12}(\mathrm{HS}, t) := (H_1(\mathrm{HS}, t), H_2(\mathrm{HS}, t))$[10]. H_{12} is internal random oracles maintained by the challenger that $\mathcal{A}$ can access to only by querying the oracle H_1, H_2. Then, the total number of queries to H_{12} is at most $q_{H_1} + q_{H_2}$. Apparently, G_3 is consistent with G_2 in $\mathcal{A}$'s view. Therefore, we have

$$\Pr[G_2^{\mathcal{A}} \Rightarrow 1] = \Pr[G_3^{\mathcal{A}} \Rightarrow 1]$$

GAME G_4: In game G_4, the random oracles G accessed by $\mathcal{A}$ is replaced by the oracles G' (which reprogram $G(K^*)$ to a random value HS_1). Note that the challenger still uses G to simulate $\mathsf{O}^{\mathrm{Dec}}$ and $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$. In particular, we delay our analysis of $|\Pr[G_3^{\mathcal{A}} \Rightarrow 1] - \Pr[G_4^{\mathcal{A}} \Rightarrow 1]|$ in Propositions 1, 2, 3.

GAME G_5: The game G_5 is the same as game G_4 except that $\mathrm{HS}^* = G(K^*)$ is replaced by $\mathrm{HS}^* \leftarrow \{0,1\}^n$. Note that games G_5 and G_6 have the same distribution. Therefore, we have $\Pr[G_5^{\mathcal{A}} \Rightarrow 1] = \Pr[G_6^{\mathcal{A}} \Rightarrow 1]$.

GAME G_6: In game G_6, we modify the oracle $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ as follows: we immediately return $\perp$ if we have $K' = K^*$ during the execution of $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$. Therefore, HS^* is independent of $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ in G_6. Define the event BAD as satisfying $K' = K^*$ and $\mathrm{MAC.Vrf}(fk_S, \mathrm{txt}, \mathrm{tag}) = \mathrm{true}$ in $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$. Note that if BAD does not occur, then G_5 and G_6 are identical. Thus, we have $|\Pr[G_5^{\mathcal{A}} \Rightarrow 1] - \Pr[G_6^{\mathcal{A}} \Rightarrow 1]| \leq \Pr[\mathrm{BAD} : G_6]$. Let us analyze $\Pr[\mathrm{BAD} : G_6]$.

When $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}(\overline{\mathrm{ct}}, \overline{n}, \mathrm{tag}, \mathrm{txt})$ satisfies $\mathrm{decaps}(\mathsf{sk}, \overline{\mathrm{ct}}) = K^*$, we reprogram the corresponding $fk_S = H_4(\mathrm{SHTS})$ to a random $\mathrm{MAC}.K$. Now, we analyze the probability of the adversary distinguishing this change. Define a new algorithm $\hat{A}$ accessing $O \in \{H_4, H'_4\}$, where $H'_4(x) = H_4(x)$ (if $x \neq \mathrm{SHTS}$) and $H'_4(\mathrm{SHTS}) = \mathrm{MAC}.K$. When $O = H_4$, $\hat{A}$ can perfectly simulate G_6 before reprogramming. When $O = H'_4$, $\hat{A}$ can perfectly simulate G_6 after reprogramming. According to the OW2H lemma, we have $|\Pr[1 \leftarrow \hat{A}^{H_4}] - \Pr[1 \leftarrow \hat{A}^{H'_4}]| \leq 2q_{H_4}\sqrt{P_{\mathcal{A}}}$, where $P_{\mathcal{A}}$ represents the probability that $\mathcal{A}$ finds SHTS. Note that $\mathrm{SHTS} = H_2(\mathrm{HS}^*, H_T(\overline{\mathrm{ct}}, \overline{n})) \neq \mathrm{SHTS}_b$ and HS^* is uniformly random and independent

[10] Note that if one wants to make queries to H_1 (or H_2) by accessing to H_{12}, he just needs to prepare a uniform superposition of all states in the output register responding to H_1 (or H_2). This technique [25,36] has been used in the proof of the Fujisaki-Okamoto transform.

from random oracle G and ct^*. Firstly, since the adversary $\mathcal{A}$ has obtained the value of $tk_S = H_D(\mathrm{SHTS})$ by querying $\mathsf{O}^{\mathrm{Dec}}$, $\mathcal{A}$ can find the corresponding SHTS by searching for a value x such that $H_D(x) = tk_S$. According to Lemma 2, the probability of adversary $\mathcal{B}$ finding SHTS is at most $8(q_{H_D}+1)^2\frac{1}{2^n}$. Otherwise, SHTS is independent of $\mathcal{A}$'s view, thus we have $P_{\mathcal{A}} = \frac{1}{2^n}$. In summary, the probability that adversary $\mathcal{A}$ can distinguish fk_S from a random MAC.K is at most $2q_{H_4}\sqrt{(8(q_{H_D}+1)^2+1)\frac{1}{2^n}} \leq 6q_{H_4}(q_{H_D}+2)2^{-n/2}$. Right now, $fk_S = H_4(\mathrm{SHTS})$ is truly random for the adversary $\mathcal{A}$. However, if the BAD event occurs, this means $\mathcal{A}$ successfully forges a valid tag. Therefore, one can construct an adversary $\mathcal{B}_1$ that breaks MAC EUF-0T security. More formally, $\mathcal{B}_1$ samples all the valid inputs as in G_6 and simulates the Dec oracle as in G_6. Then, when $\mathcal{A}$ submits $(\overline{\mathrm{ct}}, \overline{n}, \mathrm{tag}, \mathrm{txt})$ to $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$, $\mathcal{B}_1$ outputs $(\mathrm{txt}, \mathrm{tag})$ as a forgery. Thus, we have $\Pr[\mathrm{BAD} : G_6] \leq 6q_{H_4}(q_{H_D}+2)2^{-n/2} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_1)$. Therefore, we have

$$|\Pr[G_5^{\mathcal{A}} \Rightarrow 1] - \Pr[G_6^{\mathcal{A}} \Rightarrow 1]| \leq 6q_{H_4}(q_{H_D}+2)2^{-n/2} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_1)$$

GAME G_7. In game G_7, we use a new quantum random oracle $H_{123}(\mathrm{HS}, T \cup \perp)$ to simulate H_1, H_2, and H_3. Here, H_{123} is a quantum random oracle maintained internally by the challenger. Note that the challenger only needs to prepare a uniform superposition of all states in the output register responding to H_1, H_2 or H_3 to simulate H_1, H_2 and H_3 by accessing H_{123}. Then, the total number of queries to H_{123} is at most $q_{H_1} + q_{H_2} + q_{H_3}$. Therefore, this game is consistent with the previous game in $\mathcal{A}$'s view. Thus, we have

$$\Pr[G_6^{\mathcal{A}} \Rightarrow 1] = \Pr[G_7^{\mathcal{A}} \Rightarrow 1].$$

GAME G_8. In game G_8, the internal quantum random oracle H_{123} is replaced by quantum random oracle H'_{123}. This game is reprogramming H_{123} on (HS^*, t^*) with a random value $(\mathrm{CHTS}_1, \mathrm{SHTS}_1, \mathrm{dHS}_1)$, where $t^* = H_T(\mathrm{ct}^*, n^*)$. Applying One way to hiding (Lemma 3), we have $|\Pr[G_7^{\mathcal{A}} \Rightarrow 1] - \Pr[G_8^{\mathcal{A}} \Rightarrow 1]| \leq 2(q_{H_1} + q_{H_2} + q_{H_3} + 1)\sqrt{P[\mathcal{A}]}$, where $P[\mathcal{A}]$ represents the probability that $\mathcal{A}$ finds (HS^*, t^*). Note that the $\mathsf{O}^{\mathrm{Dec}}$ and $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ never query H_{123} on (HS^*, t^*) and the $\mathsf{O}^{\mathrm{Dec}}$ oracle completely hides HS^*. Note that HS^* is uniformly random and independent of the RO G, $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ in G_8, thus HS^* is independent and random from the adversary $\mathcal{A}$'s view. Thus,we have $\Pr[\mathcal{A}] \leq 1/2^n$. Therefore, we have

$$|\Pr[G_7^{\mathcal{A}} \Rightarrow 1] - \Pr[G_8^{\mathcal{A}} \Rightarrow 1]| \leq 2(q_{H_1} + q_{H_2} + q_{H_3} + 1)2^{-n/2} = 2q_{123}2^{-n/2}$$

GAME G_9. The game G_9 is the same as game G_7, except that the tuple $(\mathrm{CHTS}_0, \mathrm{SHTS}_0, \mathrm{dHS}_0) = H_{123}(\mathrm{HS}^*, t^*)$ is replaced by

$(\mathrm{CHTS}_0, \mathrm{SHTS}_0, \mathrm{dHS}_0) \leftarrow\!\$\ \{0,1\}^{3n}$, Note that games G_8 and G_9 have the same distribution. Therefore, we have

$$\Pr[G_8^{\mathcal{A}} \Rightarrow 1] = \Pr[G_9^{\mathcal{A}} \Rightarrow 1] = 1/2.$$

Now we only need to use the underlying OW-CPA/D-OW-CPA/IND-CPA secure KEM and EUF-0T secure MAC to bound the probability $|\Pr[G_3^{\mathcal{A}} \Rightarrow 1] - \Pr[G_4^{\mathcal{A}} \Rightarrow 1]|$. Specific analyses are provided in Proposition 1, Proposition 2, and Proposition 3.

Proposition 1. *There exists an adversary $\mathcal{C}$ against the OW-CPA of KEM, and adversaries $\mathcal{B}_1$, $\mathcal{B}_2$ against the EUF-0T of MAC such that $\mathcal{B}_1$, $\mathcal{B}_2$, and $\mathcal{C}$ have approximately the same running time as $\mathcal{A}$ and*

$$\begin{aligned}
&\mathrm{Adv}^{\text{IND-1CCA-MAC}^*}_{\mathrm{KEM}}(\mathcal{A}) \\
&\leq \frac{2q_{123} + 6q_{H_4}(q_{H_D}+2)}{2^{n/2}} + \frac{(q_{H_T}+4)^2 + (q_{H_2}+1)^2}{2^n} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_1) \\
&\quad + 8q_G(q_{H_1}+q_{H_2}+1)\cdot \\
&\qquad \sqrt{\mathrm{Adv}^{\text{OW-CPA}}_{\mathrm{KEM}}(\mathcal{C}) + \frac{1}{2^{2n}} + 6q_{H_4}(q_{H_D}+2)2^{-n/2} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_2)}.
\end{aligned}$$

The proof of Proposition 1. Define games $G_{1C} - G_{3C}$ as in Fig. 12,13.
GAME $G_{1\mathcal{C}}$. In this game, we use the OW2H Lemma (Lemma 3) to analyze the impact of reprogramming in G_3. Let $z_1 = (\mathsf{pk}, \mathsf{sk}, \mathrm{ct}^*, \mathrm{HS}^*, n^*, b, \mathrm{CHTS}_1, \mathrm{SHTS}_1, \mathrm{dHS}_1)$, where $(\mathsf{pk}, \mathsf{sk}) \leftarrow gen$, $(\mathrm{ct}^*, K^*) \leftarrow\!\$\ \mathrm{encaps}(\mathsf{pk})$, $b \leftarrow\!\$\ \{0,1\}$, $n^*, \mathrm{HS}^* \leftarrow\!\$\ \{0,1\}^n$, and $(\mathrm{CHTS}_1, \mathrm{SHTS}_1, \mathrm{dHS}_1) \leftarrow\!\$\ \{0,1\}^{3n}$. Sample $G \leftarrow\!\$\ \Omega_G, H_{12} \leftarrow\!\$\ \Omega_{H_{12}}, H_k \leftarrow\!\$\ \Omega_{H_k}(k = 3,4,D,T)$. Let G' be an oracle such that $G'(K^*) = \mathrm{HS}^*$. Let $\mathcal{C}'^{|O\rangle,|H_{12}\rangle,|H_k\rangle(k=3,4,D,T)}(z_1)$ $(O \in G, G')$ be an oracle algorithm that first compute $(\mathrm{CHTS}_0, \mathrm{SHTS}_0, \mathrm{dHS}_0) = (H_1(\mathrm{HS}^*, H_T(\mathrm{ct}^*, n^*)), H_2(\mathrm{HS}^*, H_T(\mathrm{ct}^*, n^*)), H_3(\mathrm{HS}^*))$, then runs $\mathcal{A}^{|O\rangle,|H_i\rangle(\mathrm{k}=1,2,3,4,D,T),\mathsf{O}^{\mathrm{Dec}},\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}}(\mathsf{pk}, \mathrm{ct}^*, n^*, \mathrm{CHTS}_b, \mathrm{SHTS}_b, \mathrm{dHS}_b)$ (the simulation of $\mathsf{O}^{\mathrm{Dec}}, \mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ is the same as G_3, G_4) to obtain b', and returns $b' =? b$. Thus, we have $\Pr[G_3^{\mathcal{A}} \Rightarrow 1] = \Pr[1 \leftarrow \mathcal{C}'^{|G'\rangle,|H_{12}\rangle,|H_k\rangle}(z_1)]$ and $\Pr[G_4^{\mathcal{A}} \Rightarrow 1] = \Pr[1 \leftarrow \mathcal{C}'^{|G\rangle,|H_{12}\rangle,|H_k\rangle}(z_1)]$.

Let $\mathcal{C}(z_1)$ be an algorithm that randomly selects $j \in [q_G - 1]$, runs $\mathcal{C}'^{|G\rangle,|H_{12}\rangle,|H_k\rangle}(z_1)$ up until (just before) the $(j+1)$-th query, measures the query input registers in the computational basis, and outputs measurement outcomes. Thus, we have that $\Pr[G_{1\mathcal{C}}^{\mathcal{A}} \Rightarrow 1] = \Pr[K^* \leftarrow \mathcal{C}^{|G\rangle,|H_{12}\rangle,|H_k\rangle}(z_1)]$. Therefore, according to Lemma 3, we have [11]

$$|\Pr[G_3^{\mathcal{A}} \Rightarrow 1] - \Pr[G_4^{\mathcal{A}} \Rightarrow 1]| \leq 2q_G\sqrt{\Pr[G_{1\mathcal{C}}^{\mathcal{A}} \Rightarrow 1]}.$$

[11] The quantum random oracles $|H_{12}\rangle, |H_k\rangle$ are independent of $|G\rangle$. Therefore, when we apply Lemma 3, we assume that $|H_{12}\rangle, |H_k\rangle$ are simulated by $\mathcal{C}'$ and that $|G\rangle$ is the only QRO it accesses.

GAMES $G_{1\mathcal{C}} - G_{3\mathcal{C}}$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \text{gen}, H_{12} \leftarrow\!\$\ \Omega_{H_{12}}, H_{k\in\{3,4,D,T\}} \leftarrow\!\$\ \Omega_{H_k}, G \leftarrow\!\$\ \Omega_G$

$(\text{ct}^*, K^*) \leftarrow\!\$\ \text{encaps}(\mathsf{pk}), n^* \leftarrow\!\$\ \{0,1\}^n$

$b, guess \leftarrow\!\$\ \{0,1\}, \Theta = (\overline{chts}, \overline{shts}) \leftarrow\!\$\ \{0,1\}^{2n}, \text{HS}^* \leftarrow\!\$\ \{0,1\}^n$

use $H_{12} = (H_1, H_2)$ to simulate H_1, H_2

$(\text{CHTS}_0, \text{SHTS}_0, \text{dHS}_0) \leftarrow (H_{12}(\text{HS}^*, H_T(\text{ct}^*, n^*)), H_3(\text{HS}^*))$

$(\text{CHTS}_1, \text{SHTS}_1, \text{dHS}_1) \leftarrow\!\$\ \{0,1\}^{3n}$

$l = 0, j \leftarrow\!\$\ [q_G - 1], (i, \hat{b}) \leftarrow\!\$\ ([q_{H_1} + q_{H_2} - 1]) \times \{0,1\} \cup \{(q_{H_1} + q_{H_2}, 0)\}$

Run $\mathcal{A}^{|G\rangle, |H_{12}\rangle, |H_k\rangle, \mathsf{O}^{\text{Dec}}, \mathsf{O}^{\text{Dec}}_{\text{MAC}}}(\mathsf{pk}, \text{ct}^*, n^*, (\text{CHTS}_b, \text{SHTS}_b, \text{dHS}_b))$
until the (j+1)-th G query $|\psi\rangle$ $\quad //G_{1\mathcal{C}}$

Run $\mathcal{A}^{|G\rangle, |H^i_{12}\rangle, |H_k\rangle, \mathsf{O}^{\text{Dec}}, \mathsf{O}^{\text{Dec}}_{\text{MAC}}}(\mathsf{pk}, \text{ct}^*, n^*, (\text{CHTS}_b, \text{SHTS}_b, \text{dHS}_b))$
until the (j+1)-th G query $|\psi\rangle$ $\quad //G_{2\mathcal{C}}, G_{3\mathcal{C}}$

$K' \leftarrow M|\psi\rangle$

return $K^* =? K'$

$\mathsf{O}^{\text{Dec}}(\overline{\text{ct}}, \overline{n})$

if more than 1 query : **return** $\perp$
if $(\overline{\text{ct}}, \overline{n}) = (\text{ct}^*, n^*)$: **return** $\perp$
if $guess = 0$: **return** $\perp$ $\quad //G_{2\mathcal{C}}-$
if $guess = 1$: $\quad //\ \text{G}_{2\mathcal{C}}-$
  return $H_D(\overline{chts}), H_D(\overline{shts})$
$\overline{K'} \leftarrow \text{decaps}(\mathsf{sk}, \overline{\text{ct}})$
if $\overline{K'} = \perp$: **return** $\perp$
if $\overline{K'} = K^*$: $\overline{\text{HS}'} = \text{HS}^*$
$\overline{\text{HS}'} \leftarrow G'(\overline{K'})$
$\overline{\text{CHTS}} \leftarrow H_1(\overline{\text{HS}'}, H_T(\overline{\text{ct}}, \overline{n}))$
$\overline{\text{SHTS}} \leftarrow H_2(\overline{\text{HS}'}, H_T(\overline{\text{ct}}, \overline{n}))$
$\overline{tk_C} \leftarrow H_D(\overline{\text{CHTS}})$
$\overline{tk_S} \leftarrow H_D(\overline{\text{SHTS}})$
return $\overline{tk_C}, \overline{tk_S}$

$\mathsf{O}^{\text{Dec}}_{\text{MAC}}(\overline{\text{ct}}, \overline{n}, \text{tag}, \text{txt})$

if more than 1 query : **return** $\perp$
if $(\overline{\text{ct}}, \overline{n}) = (\text{ct}^*, n^*)$: **return** $\perp$
if $guess = 0$: **return** $\perp$ $\quad //G_{2\mathcal{C}}-$
$\overline{K'} \leftarrow \text{decaps}(\mathsf{sk}, \overline{\text{ct}})$ $\quad //G_{1\mathcal{C}}, G_{2\mathcal{C}}$
$\overline{\text{HS}'} \leftarrow G'(\overline{K'})$ $\quad //G_{1\mathcal{C}}, G_{2\mathcal{C}}$
$\overline{\text{SHTS}} \leftarrow H_2(\overline{\text{HS}'}, H_T(\overline{\text{ct}}, \overline{n}))//G_{1\mathcal{C}}$
$\overline{fk_S} \leftarrow H_4(\overline{\text{SHTS}})$ $\quad //G_{1\mathcal{C}}$
$\overline{fk_S} \leftarrow H_4(\overline{shts})$ $\quad //G_{2\mathcal{C}}-$
if $\text{MAC.Vrf}(\overline{fk_S}, \text{txt}, \text{tag}) = \mathsf{true}$:
  return $\overline{\text{HS}'}$ $\quad //\ \text{G}_{1\mathcal{C}}, G_{2\mathcal{C}}$
  return HS_{i+1} $//\ \text{G}_{3\mathcal{C}}$
return $\perp$

Fig. 12: Games $G_{1\mathcal{C}} - G_{3\mathcal{C}}$ for the proof of Proposition 1

$H^i_{12}(\mathrm{HS}, t)$	G'
1 : **if** $l \geq (i + \hat{b}) \wedge (\mathrm{HS}, t) = (\mathrm{HS}_{i+1}, t_{i+1})$:	1 : **if** $K' = K^*$: **return** HS^*
2 : / $(\mathrm{HS}_{i+1}, t_{i+1})$ is the measurement outcome	2 : **else return** $G(K')$
3 : / on $\mathcal{A}$'s $(i+1)$-th query input register	
4 : **return** Θ	
5 : **else return** $\mathrm{H}_{12}(\mathrm{HS}, t)$	
6 : $l = l + 1$	

Fig. 13: H^i_{12}, G' for the proof of Proposition 1

GAME $G_{2\mathcal{C}}$. In game $G_{2\mathcal{C}}$, we use the refined optional-query technique (Lemma 6) to simulate the $\mathsf{O}^{\mathrm{Dec}}(\overline{\mathrm{ct}}, \overline{n})$ oracle without the secret key. Firstly, we sample $guess \leftarrow \{0,1\}$ to guess whether $\mathrm{decaps}(\mathsf{sk}, \overline{\mathrm{ct}}) = \bot$. Here, we always have a $1/2$ probability of guessing correctly. In the case of $guess = 0$ (the decaps result is $\bot$), we simply return $\bot$ in $\mathsf{O}^{\mathrm{Dec}}$ and $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$. Otherwise, we use the refined optional-query technique to simulate $\mathsf{O}^{\mathrm{Dec}}$ without the secret key. In the discussion below, we consider the case of $guess = 1$ ($\mathrm{decaps}(\mathsf{sk}, \overline{\mathrm{ct}}) \neq \bot$).

Let $E^{|H_{12}\rangle}$ be an oracle algorithm that samples $(\mathsf{pk}, \mathsf{sk}, K^*, \mathrm{ct}^*, n^*, b, \mathrm{HS}^*, j)$. Subsequently it runs $\mathcal{A}^{|\mathrm{G}\rangle, |\mathrm{H}_{12}\rangle, |\mathrm{H}_k\rangle, \mathsf{O}^{\mathrm{Dec}}, \mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}}$ as in game $G_{1\mathcal{C}}$. Let $(\overline{\mathrm{ct}}, \overline{n})$ and $(\overline{\mathrm{ct}}, \overline{n}, \mathrm{tag}, \mathrm{txt})$ be $\mathcal{A}$'s queries to the $\mathsf{O}^{\mathrm{Dec}}$ and $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ oracle. Let $\overline{K} = \mathrm{decaps}(\mathsf{sk}, \overline{\mathrm{ct}})$, $\overline{\mathrm{HS}} = G(\overline{K})$, $\bar{t} = H_T(\overline{\mathrm{ct}}, \overline{n})$, $x = (x_1, x_2) = (\overline{\mathrm{HS}}, \bar{t})$, $y = H_{12}(x) = (H_1(x), H_2(x))$, and $z = (z_1, z_2, z_3, z_4) = (\mathsf{O}^{\mathrm{Dec}}(\overline{\mathrm{ct}}, \overline{n}), \mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}(\overline{\mathrm{ct}}, \overline{n}, \mathrm{tag}, \mathrm{txt}), K^*, K')$. E outputs (x, z). Let $V_1(x, y, z) = (H_D(y) = z_1) \wedge (((x_1 = z_2) \wedge (z_2 \neq \bot)) \vee (z_2 = \bot))$, and $V_2(x, y, z) = (z_3 = z_4)$. Instantiating the predicate V in Lemma 6 by $V = V_1 \wedge V_2$. Note that in $G_{1\mathcal{C}}$ the return of the $\mathsf{O}^{\mathrm{Dec}}$ oracle is exactly $(H_D(H_1(x)), H_D(H_2(x))) = H_D(y)$ and $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ oracle returns either x_1 or $\bot$. That is, $V_1 = 1$ is always satisfied. Thus, we have $\Pr[G^{\mathcal{A}}_{1\mathcal{C}} \Rightarrow 1] = \sum_{x_0} \Pr_{H_{12}}[x = x_0 \wedge V(x, H_{12}(x), z) = 1 : (x, z) \leftarrow E^{|H_{12}\rangle}]$.

Note that E needs to implicitly query a $H_{12}(\overline{\mathrm{HS}}, \bar{t})$ to simulate the $\mathsf{O}^{\mathrm{Dec}}$, $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ oracle. That is, E makes $(q_{H_1} + q_{H_2} + 1)$ H_{12} queries in total. In the following, unless otherwise specified, the H_{12}-queries we mention do not include this implicit H_{12}-query. Let $S^E(\Theta)$ be an oracle algorithm that always returns Θ for E's implicit classical H_{12}-query $H_{12}(\overline{\mathrm{HS}}, \bar{t})$. S samples a uniform $(i, \hat{b}) \leftarrow ([q_{H_1} + q_{H_2} - 1] \times \{0,1\}) \cup \{(q_{H_1} + q_{H_2}, 0)\}$, runs $E^{|H_{12}\rangle}$ until the E's $(i+1)$-th query (excluding the implicit H_{12}-query), measures the query input registers to obtain x, continues to run $E^{|H_{12}\rangle}$ until the $(i + \hat{b} + 1)$-th H_{12}-query, reprograms H_{12} to $H_{12_{x\Theta}}$ ($H_{12_{x\Theta}}(x) = \Theta$ and $H_{12_{x\Theta}}(x') = H(x')$ for all $x' \neq x$), and runs $E^{H_{12_{x\Theta}}}$ until the end to output z. Let $x = (\overline{\mathrm{HS}}, \bar{t})$, $y = \Theta$, and $z = (z_1, z_2, z_3, z_4) = (\mathsf{O}^{\mathrm{Dec}}(\overline{\mathrm{ct}}, \overline{n}), \mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}(\overline{\mathrm{ct}}, \overline{n}, \mathrm{tag}, \mathrm{txt}), K^*, K')$. S^E outputs (x, z). Note that $V_1(x, y, z) = (H_D(y) = z_1) \wedge (((\overline{\mathrm{HS}} = z_2) \wedge (z_2 \neq \bot)) \vee (z_2 = \bot)) = 1$ for S^E. Sample $\Theta = (\overline{chts}, \overline{shts}) \leftarrow \{0,1\}^{2n}$ and $H_{12} \leftarrow \Omega_{H_{12}}$. Then, $S^E(\Theta)$

perfectly simulates game $G_{2\mathcal{C}}$ and we have $\Pr[G^{\mathcal{A}}_{2\mathcal{C}} \Rightarrow 1] = \sum x_0 \Pr_{H,\Theta}[x = x_0 \wedge V(x,\Theta,z) = 1 : (x,z) \leftarrow S^E]$.

According to Lemma 6, $\sum_{x_0} \Pr_{H_{12}}[x = x_0 \wedge V(x, H_{12}(x), z) = 1 : (x,z) \leftarrow E^{|H_{12}\rangle}] \leq 2(2q_{H_{12}}+1)^2 \sum_{x_0} \Pr_{H_{12},\Theta}[x = x_0 \wedge V(x,\Theta,z) = 1 : (x,z) \leftarrow S^E] + 8(q_{H_{12}}+1)^2 \frac{1}{2^{2n}}$. Here, $q_{H_{12}} = q_{H_1} + q_{H_2}$. Therefore, combined with the probability that the *guess* is correct, we have

$$\Pr[G^{\mathcal{A}}_{1\mathcal{C}} \Rightarrow 1] \leq 16(q_{H_1} + q_{H_2} + 1)^2 (\Pr[G^{\mathcal{A}}_{2\mathcal{C}} \Rightarrow 1] + \frac{1}{2^{2n}}).$$

GAME $G_{3\mathcal{C}}$: In game $G_{3\mathcal{C}}$, we simulate the $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ oracle without the secret key. Recall that the input $(\overline{\mathrm{ct}}, \overline{n}, \cdot, \cdot)$ to $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ is consistent with the input $(\overline{\mathrm{ct}}, \overline{n})$ to $\mathsf{O}^{\mathrm{Dec}}$. In $G_{2\mathcal{C}}$, we have made a guess of whether $\mathrm{decaps}(\mathsf{sk}, \overline{\mathrm{ct}}) = \perp$ and reprogrammed $H_{12}(\overline{\mathrm{HS}}, \overline{\mathrm{t}})$ to $\Theta = (\overline{chts}, \overline{shts})$ conditioned on $(i,b) \leftarrow\$ ([q_{H_1} + q_{H_2} - 1] \times \{0,1\}) \cup \{(q_{H_1} + q_{H_2}, 0)\}$. In game $G_{3\mathcal{C}}$, the simulation of $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ is as follows: If $guess = 0$, we just return $\perp$ in $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$. Otherwise, we compute $fk_S = H_4(\overline{shts})$, verify if $\mathrm{MAC.Vrf}(fk_S, \mathrm{txt}, \mathrm{tag})$ is correct.

- If $\mathrm{MAC.Vrf}(fk_S, \mathrm{txt}, \mathrm{tag})$ returns true and the $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ query occurs after the measurement, namely, the $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ query occurs after the $(i+1)$-th H_{12} query, we can directly return $\overline{\mathrm{HS}}$ by using the measurement value $x = (\mathrm{HS}_{i+1}, t_{i+1})$. Note that since $V_1 = 1$, this simulation is perfect.
- If $\mathrm{MAC.Vrf}(fk_S, \mathrm{txt}, \mathrm{tag})$ returns true and the $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ query occurs before the measure, namely, the $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ query occurs before the $(i+1)$-th H_{12} query, it means that the adversary $\mathcal{A}$ besides obtaining tk_S from $\mathsf{O}^{\mathrm{Dec}}$, knows nothing about $\overline{shts}$ from the quantum random oracle H_{12}, because H_{12} is independent of $\Theta = (\overline{shts}, \overline{chts})$ before the $(i+1)$-th H_{12} query . Below, we carefully analyze the probability of the second case.
  We reprogram $fk_S = H_4(\overline{shts})$ to a random value $\overline{\mathrm{MAC}.K}$. Now, we analyze the probability of the adversary distinguishing the genuine fk_S from the random value. Define a new algorithm $\hat{A}$ accessing $O \in \{H_4, H'_4\}$, where $H'_4(x) = H_4(x)$ (if $x \neq \overline{shts}$) and $H'_4(\overline{shts}) = \overline{\mathrm{MAC}.K}$. When $O = H_4$, $\hat{A}$ can perfectly simulate $G_{3\mathcal{C}}$ before reprogramming. When $O = H'_4$, $\hat{A}$ can perfectly simulate $G_{3\mathcal{C}}$ after reprogramming. According to the OW2H lemma, we have $|\Pr[1 \leftarrow \hat{A}^{H_4}] - \Pr[1 \leftarrow \hat{A}^{H'_4}]| \leq 2q_{H_4}\sqrt{P_{\mathcal{A}}}$, where $P_{\mathcal{A}}$ represents the probability that $\mathcal{A}$ finds $\overline{shts}$ access with tk_S and H_4. Firstly, since the adversary $\mathcal{A}$ has obtained the value of $tk_S = H_D(\overline{shts})$ by querying $\mathsf{O}^{\mathrm{Dec}}$, $\mathcal{A}$ can find the corresponding $\overline{shts}$ by searching for a value x such that $H_D(x) = tk_S$. According to Lemma 2, the probability of adversary $\mathcal{A}$ finding $\overline{shts}$ is at most $8(q_{H_D}+1)^2 \frac{1}{2^n}$. Otherwise, $\overline{shts}$ is independent of $\mathcal{A}$'s view, thus we have $P_{\mathcal{A}} = \frac{1}{2^n}$. Note that $\overline{shts} \neq \mathrm{SHTS}_b$ according to G_2. In summary, the probability that adversary $\mathcal{A}$ can distinguish fk_S from a random one is at most $2q_{H_4}\sqrt{(8(q_{H_D}+1)^2+1)\frac{1}{2^n}} \leq 6q_{H_4}(q_{H_D}+2)2^{-n/2}$.
  Right now, fk_S is truly random for the adversary $\mathcal{A}$. However, $\mathcal{A}$ successfully forges a valid tag in this case. Therefore, one can construct an adversary $\mathcal{B}_2$ that breaks MAC EUF-0T security. More formally, $\mathcal{B}_2$ samples all the valid

inputs as in $G_{2\mathcal{C}}$ and simulates the $\mathsf{O}^{\mathrm{Dec}}$ oracle as in $G_{2\mathcal{C}}$. Then, when $\mathcal{A}$ submits $(\overline{\mathrm{ct}}, \overline{n}, \mathrm{tag}, \mathrm{txt})$ to $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$, $\mathcal{B}$ outputs $(\mathrm{txt}, \mathrm{tag})$ as a forgery. Thus, in the second case, the probability for this case to occur is at most $6q_{H_4}(q_{H_D}+2)2^{-n/2} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_2)$.

- Otherwise, output $\perp$.

Through the analysis above, the probability of case 2 occurring is at most $6q_{H_4}(q_{H_D}+2)2^{-n/2} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B})$. We can observe that in the scenario where case 2 does not occur, we can perfectly simulate the $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ oracle without sk. Finally, from this point onward in the game, we can simulate both $\mathsf{O}^{\mathrm{Dec}}$ and $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ without sk. Therefore, we have

$$|\Pr[G^{\mathcal{A}}_{2\mathcal{C}} \Rightarrow 1] - \Pr[G^{\mathcal{A}}_{3\mathcal{C}} \Rightarrow 1]| \leq 6q_{H_4}(q_{H_D}+2)2^{-n/2} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_2).$$

Now, we can construct an OW-CPA adversary $\mathcal{C}(pk, \mathrm{ct}^*)$ against KEM, where $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathrm{gen}$, $(K^*, \mathrm{ct}^*) \leftarrow \mathrm{encaps}(pk)$. $\mathcal{C}$ samples $n^*, b, guess, guess', \Theta, \mathrm{HS}^*, \mathrm{CHTS}_1, \mathrm{SHTS}_1, \mathrm{dHS}_1, j, i, \hat{b}$ as in game $G_{3\mathcal{C}}$ and picks five $2q_{H_k}$-wise $(k \in \{12, 3, 4, H, T\})$ independent functions and a $2q_G$-wise independent functions (indistinguishable from a random function for a $q_{H_k}(q_G)$-query adversary according to Lemma 1.) And $\mathcal{C}$ runs $\mathcal{A}^{G, |\mathrm{H}_{12}\rangle, |\mathrm{H}_k\rangle (k=3,4,D,T), \mathsf{O}^{\mathrm{Dec}}, \mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}}(\mathsf{pk}, \mathrm{ct}^*, n^*, \mathrm{CHTS}^*_b, \mathrm{SHTS}^*_b, \mathrm{dHS}^*_b)$ (the simulations of G, H_{12}, H_k, $\mathsf{O}^{\mathrm{Dec}}$, $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ are the same as the ones in game $G_{3\mathcal{C}}$) until the $(j+1)$-th query, measure $\mathcal{A}$'s query input register to obtain K', and finally output K' as a return. It is obvious that the advantage of $\mathcal{C}$ against the OW-CPA security of KEM is exactly $\Pr[G^{\mathcal{A}}_{3\mathcal{C}} \Rightarrow 1]$. Putting everything together, we have

$$\begin{aligned}
&\mathrm{Adv}^{\text{IND-1CCA-MAC}^*}_{\mathrm{KEM}}(\mathcal{A}) \\
&\leq \frac{2q_{123} + 6q_{H_4}(q_{H_D}+2)}{2^{n/2}} + \frac{(q_{H_T}+4)^2 + (q_{H_2}+1)^2}{2^n} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_1) \\
&\quad + 8q_G(q_{H_1} + q_{H_2} + 1) \cdot \\
&\qquad \sqrt{\mathrm{Adv}^{\text{OW-CPA}}_{\mathrm{KEM}}(\mathcal{C}) + \frac{1}{2^{2n}} + 6q_{H_4}(q_{H_D}+2)2^{-n/2} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_2)}.
\end{aligned}$$

□

Proposition 2. *There exists an adversary $\hat{\mathcal{C}}$ against the OW-CPA of DKEM* [12] *and adversaries $\mathcal{B}_1$, $\mathcal{B}_2$ against the EUF-0T of MAC such that $\mathcal{B}_1$, $\mathcal{B}_2$ $\mathcal{C}$ have*

[12] As the simulation of $\mathsf{O}^{\mathrm{Dec}}$ and $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ is the same as in Proposition 1, we do not rely on the rigid property of DPKE here.

approximately the same running time as $\mathcal{A}$ and

$$\begin{aligned}
&\mathrm{Adv}^{\text{IND-1CCA-MAC}^*}_{\mathrm{KEM}}(\mathcal{A})\\
&\le \frac{2q_{123}+6q_{H_4}(q_{H_D}+2)}{2^{n/2}} + \frac{(q_{H_T}+4)^2+(q_{H_2}+1)^2}{2^n} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_1)\\
&\quad + 8(q_{H_1}+q_{H_2}+1)\cdot\\
&\qquad \sqrt{\mathrm{Adv}^{\text{OW-CPA}}_{\mathrm{DKEM}}(\hat{\mathcal{C}}) + \frac{1}{2^{2n}} + \delta + 6q_{H_4}(q_{H_D}+2)2^{-n/2} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_2)}.
\end{aligned}$$

The proof of Proposition 2. Define games $G_{1\hat{\mathcal{C}}} - G_{4\hat{\mathcal{C}}}$ as in Fig. 14,15.
GAME $G_{1\hat{\mathcal{C}}}$. In this game, we use the Double-sided OW2H Lemma (Lemma 4) to analyze the impact of reprogramming in G_4. Let $z_1 = (\mathsf{pk}, \mathsf{sk}, \mathrm{ct}^*, \mathrm{HS}^*, n^*, b, \mathrm{CHTS}_1, \mathrm{SHTS}_1, \mathrm{dHS}_1)$, where $(\mathsf{pk}, \mathsf{sk}) \leftarrow gen$, $(\mathrm{ct}^*, K^*) \leftarrow\!\$\ \mathrm{encaps}(\mathsf{pk})$, $b \leftarrow\!\$\ \{0,1\}$, $n^*, \mathrm{HS}^* \leftarrow\!\$\ \{0,1\}^n$, and $(\mathrm{CHTS}_1, \mathrm{SHTS}_1, \mathrm{dHS}_1) \leftarrow\!\$\ \{0,1\}^{3n}$. Sample $G \leftarrow\!\$\ \Omega_G, H_{12} \leftarrow\!\$\ \Omega_{H_{12}}, H_k \leftarrow\!\$\ \Omega_{H_k} (k = 3,4,D,T)$. Let G' be an oracle such that $G'(K^*) = \mathrm{HS}^*$. Let $\hat{\mathcal{C}}'^{|O\rangle,|H_{12}\rangle,|H_k\rangle(k=3,4,D,T)}(z_1)$ $(O \in G, G')$ be an oracle algorithm that first computes $(\mathrm{CHTS}_0, \mathrm{SHTS}_0, \mathrm{dHS}_0) = (H_1(\mathrm{HS}^*, H_T(\mathrm{ct}^*, n^*)), H_2(\mathrm{HS}^*, H_T(\mathrm{ct}^*, n^*)), H_3(\mathrm{HS}^*))$, then runs $\mathcal{A}^{|O\rangle,|H_i\rangle(\mathrm{k}=1,2,3,4,D,T),\mathsf{O}^{\mathrm{Dec}},\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}}(\mathsf{pk}, \mathrm{ct}^*, n^*, \mathrm{CHTS}_b, \mathrm{SHTS}_b, \mathrm{dHS}_b)$[13] to obtain b', and returns $b' =?b$. Thus, we have $\Pr[G_3^{\mathcal{A}} \Rightarrow 1] = \Pr[1 \leftarrow \hat{\mathcal{C}}'^{|G'\rangle,|H_{12}\rangle,|H_k\rangle}(z_1)]$ and $\Pr[G_4^{\mathcal{A}} \Rightarrow 1] = \Pr[1 \leftarrow \hat{\mathcal{C}}'^{|G\rangle,|H_{12}\rangle,|H_k\rangle}(z_1)]$.

Lemma 4 states that [14]there exists an oracle algorithm $\hat{\mathcal{C}}^{|G\rangle,|G'\rangle,|H_{12}\rangle,H_k\rangle}(z_1)$ such that: $|\Pr[1 \leftarrow \hat{\mathcal{C}}'^{|G\rangle,|H_{12}\rangle,|H_k\rangle}(z_1)] - \Pr[1 \leftarrow \hat{\mathcal{C}}'^{|G'\rangle,|H_{12}\rangle,|H_k\rangle}(z_1)]| \le 2\sqrt{\Pr[K_0^* \leftarrow \hat{\mathcal{C}}_1^{|G\rangle,|G'\rangle,|H_{12}\rangle,|H_k\rangle}(z_1)]} = 2\sqrt{\Pr[G_{1\hat{\mathcal{C}}} \Rightarrow 1]}$.

GAME $G_{2\hat{\mathcal{C}}}$: In game $G_{2\hat{\mathcal{C}}}$, we change the computation process of G'. In game $G_{2\hat{\mathcal{C}}}$, the judgement condition $K = K^*$ is replaced by $\mathrm{ct}^* = \mathrm{enc}'(\mathsf{pk}, K)$ without knowledge of K_0^*. Define COLL as an event that there is a $K \neq K^*$ such that $\mathrm{enc}'(\mathsf{pk}, K) = \mathrm{ct}^* = \mathrm{enc}'(\mathsf{pk}, K^*)$. Note that if COLL does not happen (implied by the δ-correctnes), then GAME $G_{1\hat{\mathcal{C}}}$ and GAME $G_{2\hat{\mathcal{C}}}$ have the same distribution. Thus, we have

$$|\Pr[G_{1\hat{\mathcal{C}}}^{\mathcal{A}} \Rightarrow 1] - \Pr[G_{2\hat{\mathcal{C}}}^{\mathcal{A}} \Rightarrow 1]| \le \delta$$

GAME $G_{3\hat{\mathcal{C}}}$. In game $G_{3\hat{\mathcal{C}}}$, the $\mathsf{O}^{\mathrm{Dec}}$, $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ oracle is modified in the same way as in Proposition 1, $G_{2\mathcal{C}}$. That is, we make a random guess to determine whether $\mathrm{decaps}(\mathsf{sk}, \overline{\mathrm{ct}}) = \perp$ we reprogram H_{12} conditioned on $(i, \hat{b}) \leftarrow\!\$\ ([q_{H_1} + q_{H_2} - 1] \times$

[13] $\mathsf{O}^{\mathrm{Dec}}$ and $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ are internally computed by $\hat{\mathcal{C}}'$ based on G_3 and G_4. For a clear presentation, in Fig.14, we use external $\mathsf{O}^{\mathrm{Dec}}$ and $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ oracles for simulation (implemented as in G_3 and G_4).

[14] The quantum random oracles $|H_{12}\rangle, |H_k\rangle$ are independent of $|G\rangle$. Therefore, when we apply Lemma 4, we assume that $|H_{12}\rangle, |H_k\rangle$ are simulated by $\mathcal{C}'$ and that $|G\rangle$ is the only QRO it accesses.

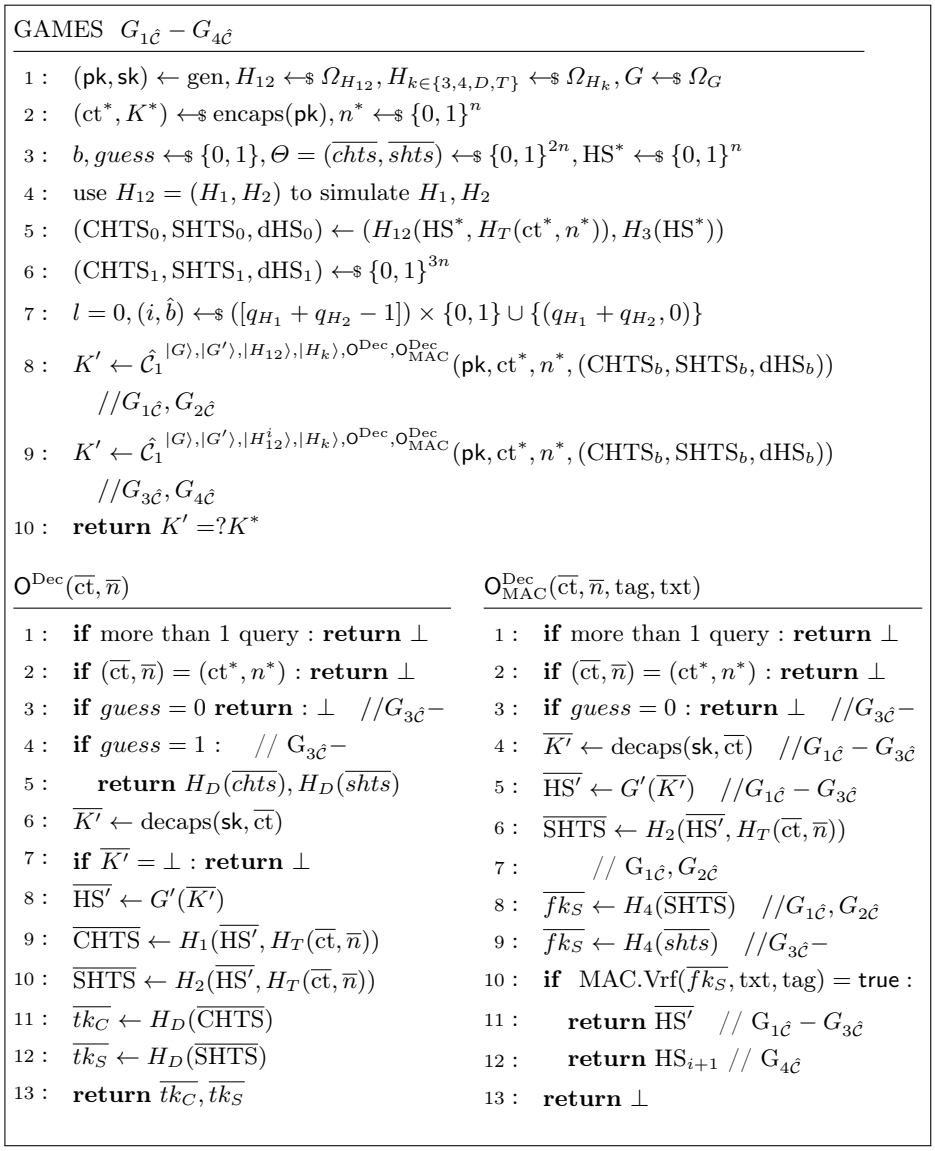

Fig. 14: Games $G_{1\hat{\mathcal{C}}} - G_{4\hat{\mathcal{C}}}$ for the proof of Proposition 2

$H^i_{12}(\mathrm{HS}, t)$	$G'(K)$
1 : **if** $l \geq (i + \hat{b}) \wedge (\mathrm{HS}, t) = (\mathrm{HS}_{i+1}, t_{i+1})$:	1 : **if** $K = K^*$: // $\mathrm{G}_{1\hat{\mathcal{C}}}$
2 : / $(\mathrm{HS}_{i+1}, t_{i+1})$ is the measurement outcome	2 : **if** $\mathrm{enc}'(\mathsf{pk}, K) = \mathrm{ct}^*$:
3 : / on $\mathcal{A}$'s $(i+1)$-th query input register	3 : // $\mathrm{G}_{2\hat{\mathcal{C}}} - G_{4\hat{\mathcal{C}}}$
4 : **return** Θ	4 : **return** HS^*
5 : **else return** $\mathrm{H}_{12}(\mathrm{HS}, t)$	5 : **return** $G(K)$
6 : $l = l + 1$	

Fig. 15: H^i_{12} and G' for the proof of Proposition 2

$\{0,1\}) \cup \{(q_{H_1} + q_{H_2}, 0)\}$ and return $H_D(\overline{chts}), H_D(\overline{shts})$ in the $\mathsf{O}^{\mathrm{Dec}}$ oracle in the case of decaps$(\mathsf{sk}, \overline{\mathrm{ct}}) \neq \perp$. Note that we also compute $\overline{fk_S} = H_4(\overline{shts})$ in $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ oracle in the case of decaps$(\mathsf{sk}, \overline{\mathrm{ct}}) \neq \perp$. Using Lemma 6 in the same way as Proposition 1. Then we have

$$\Pr[G^{\mathcal{A}}_{2\hat{\mathcal{C}}} \Rightarrow 1] \leq 16(q_{H_1} + q_{H_2} + 1)^2 (\Pr[G^{\mathcal{A}}_{3\hat{\mathcal{C}}} \Rightarrow 1] + \frac{1}{2^{2n}}).$$

GAME $G_{4\hat{\mathcal{C}}}$: In game $G_{4\hat{\mathcal{C}}}$, we simulate $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ without sk in the same way as in Proposition 1, $G_{3\mathcal{C}}$. Therefore, we have

$$|\Pr[G^{\mathcal{A}}_{3\hat{\mathcal{C}}} \Rightarrow 1] - \Pr[G^{\mathcal{A}}_{4\hat{\mathcal{C}}} \Rightarrow 1]| \leq 6q_{H_4}(q_{H_D} + 2)2^{-n/2} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_2).$$

From this point onwards in this game, we can simulate both $\mathsf{O}^{\mathrm{Dec}}$ and $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ without sk.

Now, we can construct an OW-CPA adversary $\hat{\mathcal{C}}(\mathsf{pk}, ct^*)$ against DKEM, where $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathrm{gen}$, $(K^*, \mathrm{ct}^*) \leftarrow\!\!\$\ \mathrm{encaps}(pk)$. $\hat{\mathcal{C}}$ samples $\mathrm{HS}^*, b, guess, guess', \Theta, n^*, \mathrm{CHTS}_1, \mathrm{SHTS}_1, \mathrm{dHS}_1, i, \hat{b}$ as in game $G_{4\hat{\mathcal{C}}}$, and picks five $2q_{H_k}$-wise ($k \in \{12, 3, 4, H, T\}$) independent functions and a $2q_G$-wise independent functions (indistinguishable from a random function for a $q_{H_k}(q_G)$-query adversary according to Lemma 1.) And $\hat{\mathcal{C}}$ runs $\hat{\mathcal{C}}_1^{|G\rangle, |G'\rangle, |\mathrm{H}_{12}\rangle, |\mathrm{H}_k\rangle, \mathsf{O}^{\mathrm{Dec}}, \mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}}(\mathsf{pk}, \mathrm{ct}^*, n^*, \mathrm{CHTS}^*_b, \mathrm{SHTS}^*_b, \mathrm{dHS}^*_b)$ (the simulations of the game are the same as $G_{4\hat{\mathcal{C}}}$). It is obvious that the advantage of $\hat{\mathcal{C}}$ against the OW-CPA security of DKEM is exactly $\Pr[G^{\mathcal{A}}_{4\hat{\mathcal{C}}} \Rightarrow 1]$

Putting everything together, we have

$$\begin{aligned}
&\mathrm{Adv}^{\text{IND-1CCA-MAC}^*}_{\mathrm{KEM}}(\mathcal{A}) \\
&\leq \frac{2q_{123} + 6q_{H_4}(q_{H_D} + 2)}{2^{n/2}} + \frac{(q_{H_T} + 4)^2 + (q_{H_2} + 1)^2}{2^n} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_1) \\
&\quad + 8(q_{H_1} + q_{H_2} + 1) \cdot \\
&\qquad \sqrt{\mathrm{Adv}^{\text{OW-CPA}}_{\mathrm{DKEM}}(\hat{\mathcal{C}}) + \frac{1}{2^{2n}} + \delta + 6q_{H_4}(q_{H_D} + 2)2^{-n/2} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_2)}.
\end{aligned}$$

Proposition 3. *There exists an adversary $\mathcal{D}$ against the IND-CPA of KEM and adversaries $\mathcal{B}_1$, $\mathcal{B}_2$ against the EUF-0T of MAC such that $\mathcal{B}_1$, $\mathcal{B}_2$, and $\mathcal{D}$ have approximately the same running time as $\mathcal{A}$ and*

$$\begin{aligned}
&\mathrm{Adv}^{\text{IND-1CCA-MAC}^*}_{\mathrm{KEM}}(\mathcal{A}) \\
&\leq \frac{2q_{123}+6q_{H_4}(q_{H_D}+2)}{2^{n/2}} + \frac{(q_{H_T}+4)^2+(q_{H_2}+1)^2}{2^n} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_1) \\
&\quad + 8(q_{H_1}+q_{H_2}+1)\cdot \\
&\qquad \sqrt{2\mathrm{Adv}^{\text{IND-CPA}}_{\mathrm{KEM}}(\mathcal{D}) + \frac{(q_G+1)^2}{|\mathcal{K}|} + 6q_{H_4}(q_{H_D}+3)2^{-n/2} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_2)}.
\end{aligned}$$

The proof of Proposition 3. Define games $G_{1\mathcal{D}} - G_{5\mathcal{D}}$ as in Fig. 16,17.
GAME $G_{1\mathcal{D}}$. In this game, we use the Double-sided OW2H Lemma (Lemma 4) to analyze the impact of reprogramming in G_4. Let $z_1 = (\mathsf{pk}, \mathsf{sk}, \mathrm{ct}^*, K_0^*, \mathrm{HS}^*, n^*, b, \mathrm{CHTS}_1, \mathrm{SHTS}_1, \mathrm{dHS}_1)$, where $(\mathsf{pk}, \mathsf{sk}) \leftarrow gen$, $(\mathrm{ct}^*, K_0^*) \leftarrow\!\!\$\ \mathrm{encaps}(\mathsf{pk})$, $b \leftarrow\!\!\$\ \{0,1\}$, $n^*, \mathrm{HS}^* \leftarrow\!\!\$\ \{0,1\}^n$, and $(\mathrm{CHTS}_1, \mathrm{SHTS}_1, \mathrm{dHS}_1) \leftarrow\!\!\$\ \{0,1\}^{3n}$. Sample $G \leftarrow\!\!\$\ \Omega_G, H_{12} \leftarrow\!\!\$\ \Omega_{H_{12}}, H_k \leftarrow\!\!\$\ \Omega_{H_k} (k = 3,4,D,T)$. Let G' be an oracle such that $G'(K_0^*) = \mathrm{HS}^*$. Let $\mathcal{D}'^{|O\rangle,|H_{12}\rangle,|H_k\rangle(k=3,4,D,T)}(z_1)$ $(O \in G, G')$ be an oracle algorithm that first compute $(\mathrm{CHTS}_0, \mathrm{SHTS}_0, \mathrm{dHS}_0) = (H_1(\mathrm{HS}^*, H_T(\mathrm{ct}^*, n^*)), H_2(\mathrm{HS}^*, H_T(\mathrm{ct}^*, n^*)), H_3(\mathrm{HS}^*))$, then runs $\mathcal{A}^{|O\rangle,|H_k\rangle(i=1,2,3,4,D,T),\mathsf{O}^{\mathrm{Dec}},\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}}(\mathsf{pk}, \mathrm{ct}^*, n^*, \mathrm{CHTS}_b^*, \mathrm{SHTS}_b^*, \mathrm{dHS}_b^*)$[15] to obtain b', and returns $b' =?b$. Thus, we have $\Pr[G_3^{\mathcal{A}} \Rightarrow 1] = \Pr[1 \leftarrow \mathcal{D}'^{|G'\rangle,|H_{12}\rangle,|H_k\rangle}(z_1)]$ and $\Pr[G_4^{\mathcal{A}} \Rightarrow 1] = \Pr[1 \leftarrow \mathcal{D}'^{|G\rangle,|H_{12}\rangle,|H_k\rangle}(z_1)]$.

Lemma 4 states that [16]there exists an oracle algorithm $\hat{\mathcal{D}}^{|G\rangle,|G'\rangle,|H_{12}\rangle,|H_k\rangle}(z_1)$ such that: $|\Pr[1 \leftarrow \mathcal{D}'^{|G'\rangle,|H_{12}\rangle,|H_k\rangle}(z_1)] - \Pr[1 \leftarrow \mathcal{D}'^{|G\rangle,|H_{12}\rangle,|H_k\rangle}(z_1)]| \leq 2\sqrt{\Pr[K_0^* \leftarrow \hat{\mathcal{D}}^{|G\rangle,|G'\rangle,|H_{12}\rangle,|H_k\rangle}(z_1)]} = 2\sqrt{\Pr[G_{1\mathcal{D}} \Rightarrow 1]}$.

GAME $G_{2\mathcal{D}}$. In game $G_{2\mathcal{D}}$, the $\mathsf{O}^{\mathrm{Dec}}$, $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ oracle is modified in the same way as in Proposition 1, $G_{2\mathcal{C}}$. That is, we make a random guess to determine whether $\mathrm{decaps}(\mathsf{sk}, \overline{\mathrm{ct}}) = \perp$ we reprogram H_{12} conditioned on $(i, \hat{b}) \leftarrow\!\!\$\ ([q_{H_1}+q_{H_2}-1] \times \{0,1\}) \cup \{(q_{H_1}+q_{H_2}, 0)\}$ and return $H_D(\overline{chts}), H_D(\overline{shts})$ in the $\mathsf{O}^{\mathrm{Dec}}$ oracle in the case of $\mathrm{decaps}(\mathsf{sk}, \overline{\mathrm{ct}}) \neq \perp$. Note that we also compute $\overline{fk_S} = H_4(\overline{shts})$ in $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ oracle in the case of $\mathrm{decaps}(\mathsf{sk}, \overline{\mathrm{ct}}) \neq \perp$. Using Lemma 6 in the same way as Proposition 1. Then we have

$$\Pr[G_{1\mathcal{D}}^{\mathcal{A}} \Rightarrow 1] \leq 16(q_{H_1}+q_{H_2}+1)^2(\Pr[G_{2\mathcal{D}}^{\mathcal{A}} \Rightarrow 1] + \frac{1}{2^{2n}}).$$

[15] $\mathsf{O}^{\mathrm{Dec}}$ and $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ are internally computed by $\mathcal{D}'$ based on G_3 and G_4. For a clear presentation, in Fig.14, we use external $\mathsf{O}^{\mathrm{Dec}}$ and $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ oracles for simulation (implemented as in G_3 and G_4).

[16] The quantum random oracles $|H_{12}\rangle, |H_k\rangle$ are independent of $|G\rangle$. Therefore, when we apply Lemma 4, we assume that $|H_{12}\rangle, |H_k\rangle$ are simulated by $\mathcal{C}'$ and that $|G\rangle$ is the only QRO it accesses.

GAMES $G_{1\mathcal{D}} - G_{5\mathcal{D}}$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \text{gen}, H_{12} \leftarrow\!\$\ \Omega_{H_{12}}, H_{k\in\{3,4,D,T\}} \leftarrow\!\$\ \Omega_{H_k}, G \leftarrow\!\$\ \Omega_G$

$(\text{ct}^*, K_0^*) \leftarrow\!\$\ \text{encaps}(\mathsf{pk}), K_1^* \leftarrow\!\$\ \mathcal{K}, \text{HS}^* \leftarrow\!\$\ \{0,1\}^n, \overline{b} \leftarrow\!\$\ \{0,1\}$

$b, guess \leftarrow\!\$\ \{0,1\}, \Theta = (\overline{chts}, \overline{shts}) \leftarrow\!\$\ \{0,1\}^{2n}, n^* \leftarrow\!\$\ \{0,1\}^n$

use $H_{12} = (H_1, H_2)$ to simulate H_1, H_2

$(\text{CHTS}_0, \text{SHTS}_0, \text{dHS}_0) \leftarrow (H_{12}(\text{HS}^*, H_T(\text{ct}^*, n^*)), H_3(\text{HS}^*))$

$(\text{CHTS}_1, \text{SHTS}_1, \text{dHS}_1) \leftarrow\!\$\ \{0,1\}^{3n}$

$l = 0, (i, \hat{b}) \leftarrow\!\$\ ([q_{H_1} + q_{H_2} - 1]) \times \{0,1\} \cup \{(q_{H_1} + q_{H_2}, 0)\}$

$K' \leftarrow \hat{\mathcal{D}}^{|G\rangle, |G'\rangle, |H_{12}\rangle, |H_k\rangle, \mathsf{O}^{\text{Dec}}, \mathsf{O}^{\text{Dec}}_{\text{MAC}}}(\mathsf{pk}, \text{ct}^*, n^*, (\text{CHTS}_b, \text{SHTS}_b, \text{dHS}_b))$ $//G_{1\mathcal{D}}$

$K' \leftarrow \hat{\mathcal{D}}^{|G\rangle, |G'\rangle, |H_{12}^i\rangle, |H_k\rangle, \mathsf{O}^{\text{Dec}}, \mathsf{O}^{\text{Dec}}_{\text{MAC}}}(\mathsf{pk}, \text{ct}^*, n^*, (\text{CHTS}_b, \text{SHTS}_b, \text{dHS}_b))$ $//G_{2\mathcal{D}} - G_{5\mathcal{D}}$

return $K' =? K_0^*$ $//G_{1\mathcal{D}} - G_{3\mathcal{D}}$

return $K' =? K_1^*$ $//G_{4\mathcal{D}}$

if $K' = K_{\overline{b}}^*$ **then** $\overline{b}' = 0$ **else then** $\overline{b}' = 1$ $//G_{5\mathcal{D}}$

return $\overline{b}' =? \overline{b}$ $//G_{5\mathcal{D}}$

$\mathsf{O}^{\text{Dec}}(\overline{\text{ct}}, \overline{n})$

if more than 1 query : **return** $\perp$

if $(\overline{\text{ct}}, \overline{n}) = (\text{ct}^*, n^*)$: **return** $\perp$

if $guess = 0$: **return** $\perp$ $//G_{2\mathcal{D}}-$

if $guess = 1$: // $\text{G}_{2\mathcal{D}}-$

  return $H_D(\overline{chts}), H_D(\overline{shts})$

$\overline{K'} \leftarrow \text{decaps}(\mathsf{sk}, \overline{\text{ct}})$

if $\overline{K'} = \perp$: **return** $\perp$

$\overline{\text{HS}'} \leftarrow G'(\overline{K'})$

$\overline{\text{CHTS}} \leftarrow H_1(\overline{\text{HS}'}, H_T(\overline{\text{ct}}, \overline{n}))$

$\overline{SHTS} \leftarrow H_2(\overline{\text{HS}'}, H_T(\overline{\text{ct}}, \overline{n}))$

$\overline{tk_C} \leftarrow H_D(\overline{\text{CHTS}})$

$\overline{tk_S} \leftarrow H_D(\overline{\text{SHTS}})$

return $\overline{tk_C}, \overline{tk_S}$

$\mathsf{O}^{\text{Dec}}_{\text{MAC}}(\overline{\text{ct}}, \overline{n}, \text{tag}, \text{txt})$

if more than 1 query : **return** $\perp$

if $(\overline{\text{ct}}, \overline{n}) = (\text{ct}^*, n^*)$: **return** $\perp$

if $guess = 0$: **return** $\perp$ $//G_{2\mathcal{D}}-$

$\overline{K'} \leftarrow \text{decaps}(\mathsf{sk}, \overline{\text{ct}})$ $//G_{1\mathcal{D}}, G_{2\mathcal{D}}$

$\overline{\text{HS}'} \leftarrow G'(\overline{K'})$ $//G_{1\mathcal{D}}, G_{2\mathcal{D}}$

$\overline{\text{SHTS}} \leftarrow H_2(\overline{\text{HS}'}, H_T(\overline{\text{ct}}, \overline{n}))//G_{1\mathcal{D}}$

$\overline{fk_S} \leftarrow H_4(\overline{\text{SHTS}})$ $//G_{1\mathcal{D}}$

$\overline{fk_S} \leftarrow H_4(\overline{shts})$ $//G_{2\mathcal{D}}-$

if $\text{MAC.Vrf}(\overline{fk_S}, \text{txt}, \text{tag}) = \mathsf{true}$:

  return $\overline{\text{HS}'}$ // $\text{G}_{1\mathcal{D}}, G_{2\mathcal{D}}$

  return HS_{i+1} // $\text{G}_{3\mathcal{D}}$

return $\perp$

Fig. 16: Games $G_{1\mathcal{D}} - G_{5\mathcal{D}}$ for the proof of Proposition 3

$H^i_{12}(\mathrm{HS}, t)$	$G'(K)$
1 : **if** $l \geq (i+\hat{b}) \wedge (\mathrm{HS}, t) = (\mathrm{HS}_{i+1}, t_{i+1})$:	1 : **if** $K = K_1^*$: // $\mathrm{G}_{4\mathcal{D}}$
2 : / $(\mathrm{HS}_{i+1}, t_{i+1})$ is the measurement outcome	2 : **if** $K = K_{\bar{b}}^*$: // $\mathrm{G}_{5\mathcal{D}}$
3 : / on $\mathcal{A}$'s $(i+1)$-th query input register	3 : **if** $K = K_0^*$: // $\mathrm{G}_{1\mathcal{D}} - G_{3\mathcal{D}}$
4 : **return** Θ	4 : **return** HS^*
5 : **else return** $\mathrm{H}_{12}(\mathrm{HS}, t)$	5 : **return** G(K)
6 : $l = l + 1$	

Fig. 17: H^i_{12} and G' for the proof of Proposition 3

GAME $G_{3\mathcal{D}}$: In game $G_{3\mathcal{D}}$, we simulate $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ without sk in the same way as in Proposition 1, $G_{3\mathcal{C}}$. Thus, we have

$$|\Pr[G^{\mathcal{A}}_{2\mathcal{D}} \Rightarrow 1] - \Pr[G^{\mathcal{A}}_{3\mathcal{D}} \Rightarrow 1]| \leq 6q_{H_4}(q_{H_D} + 2)2^{-n/2} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_2).$$

From this point onwards in this game, we can simulate both $\mathsf{O}^{\mathrm{Dec}}$ and $\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}$ without sk.

GAME $G_{4\mathcal{D}}$: In game $G_{4\mathcal{D}}$, we change the computation process of G' by replacing if $G(K_0^*) = K$ with if $G(K_1^*) = K$, and correspondingly replacing $K_0^* =?K'$ with $K_1^* =?K'$. Note that K_1^* is independent of $\mathsf{pk}, \mathrm{ct}^*, \mathrm{HS}^*, G$. Thus, according to lemma 5, we have

$$|\Pr[G^{\mathcal{A}}_{4\mathcal{D}} \Rightarrow 1]| \leq \frac{(q_G+1)^2}{|\mathcal{K}|}$$

GAME $G_{5\mathcal{D}}$: In game $G_{5\mathcal{D}}$, we change the computation process of G' by replacing if $G(K_1^*) = K$ with if $G(K_{\bar{b}}^*) = K$, and correspondingly replacing $K_1^* =?K'$ with: if $K_{\bar{b}}^* = K'$, set $\bar{b}' = 0$, else set $\bar{b}' = 1$, return $\bar{b}' =?\bar{b}$. Thus,

$$\begin{aligned}
\Pr[G^{\mathcal{A}}_{5\mathcal{D}} \Rightarrow 1] &= \frac{1}{2}\Pr[(K_{\bar{b}}^* = K')|\bar{b} = 0] + \frac{1}{2}\Pr[(K_{\bar{b}}^* \neq K')|\bar{b} = 1] \\
&= \frac{1}{2}\Pr[(K_{\bar{b}}^* = K')|\bar{b} = 0] + \frac{1}{2} - \frac{1}{2}\Pr[(K_{\bar{b}}^* = K')|\bar{b} = 1] \\
&= \frac{1}{2} + \frac{1}{2}(\Pr[(K_{\bar{b}}^* = K')|\bar{b} = 0] - \Pr[(K_{\bar{b}}^* = K')|\bar{b} = 1]) \\
&= \frac{1}{2} + \frac{1}{2}(\Pr[G^{\mathcal{A}}_{3\mathcal{D}} \Rightarrow 1] - \Pr[G^{\mathcal{A}}_{4\mathcal{D}} \Rightarrow 1])
\end{aligned}$$

Now, we can construct an IND-CPA adversary $\mathcal{D}(\mathsf{pk}, \mathrm{ct}^*, K_{\bar{b}}^*)$ against KEM, where $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathrm{gen}$, $(K_0^*, \mathrm{ct}^*) \leftarrow\!\$\ \mathrm{encaps}(\mathsf{pk})$, $\bar{b} \leftarrow\!\$\ \{0,1\}$, $K_1^* \leftarrow\!\$\ \mathcal{K}$. $\mathcal{D}$ samples $\mathrm{HS}^*, b, guess, guess', \Theta, n^*, \mathrm{CHTS}_1, \mathrm{SHTS}_1, \mathrm{dHS}_1, i, \hat{b}$ as in game $G_{5\hat{\mathcal{D}}}$, picks five $2q_{H_k}$-wise ($k \in \{12, 3, 4, H, T\}$) independent functions and a $2q_G$-wise independent functions (indistinguishable from a random function for a $q_{H_k}(q_G)$-query adversary according to Lemma 1.) And $\mathcal{D}$ runs

$\hat{\mathcal{D}}^{|G\rangle,|G'\rangle|\mathrm{H}_{12}\rangle,|\mathrm{H}_k\rangle,\mathsf{O}^{\mathrm{Dec}},\mathsf{O}^{\mathrm{Dec}}_{\mathrm{MAC}}}(pk, \mathrm{ct}^*, n^*, \mathrm{CHTS}^*_b, \mathrm{SHTS}^*_b, \mathrm{dHS}^*_b)$ (the simulations of the game are the same as $G_{5\mathcal{D}}$). If $K^*_{\overline{b}} = K'$, set $\overline{b}^{'} = 0$, else set $\overline{b}^{'} = 1$, return $\overline{b}^{'} = ?\overline{b}$. Thus, we have

$$|\Pr[G^{\mathcal{A}}_{5\mathcal{D}} \Rightarrow 1] - 1/2| = \mathrm{Adv}^{\text{IND-CPA}}_{\mathrm{KEM}}(\mathcal{D})$$

Putting everything together, we have

$$\begin{aligned}
&\mathrm{Adv}^{\text{IND-1CCA-MAC}^*}_{\mathrm{KEM}}(\mathcal{A}) \\
&\leq \frac{2q_{123} + 6q_{H_4}(q_{H_D} + 2)}{2^{n/2}} + \frac{(q_{H_T} + 4)^2 + (q_{H_2} + 1)^2}{2^n} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_1) \\
&\quad + 8(q_{H_1} + q_{H_2} + 1)\cdot \\
&\qquad \sqrt{2\mathrm{Adv}^{\text{IND-CPA}}_{\mathrm{KEM}}(\mathcal{D}) + \frac{(q_G + 1)^2}{|\mathcal{K}|} + 6q_{H_4}(q_{H_D} + 3)2^{-n/2} + \mathrm{Adv}^{\text{EUF-0T}}_{\mathrm{MAC}}(\mathcal{B}_2)}.
\end{aligned}$$

□

B Supporting Material: MultiStage security model for TLS 1.3

Following the work of [17, 22], we introduce the concept of multi-stage security as defined by [17]. We refer the reader to the paper [17] for more details and discussion.

Protocol-specific properties are denoted by a vector (M, AUTH, FS, USE, REPLAY), encoding: the number of stages in the protocol (M), the stage at which a key undergoes unilateral or mutual authentication (AUTH), the keys that are forward secret (FS), the intended use of keys (internal or external) within the protocol (USE), and the stage susceptible to replay attacks (REPLAY).

Then, we define the set of identities (or users) as $\mathcal{U}$, and each session is represented as $\mathsf{label} = (U, V, n) \in \mathsf{LABELS} = \mathcal{U} \times \mathcal{U} \times \mathcal{N}$. This notation denotes the n-th session of the user U with the intended partner V. In the public-key variant of the model (pMSKE), each identity U is associated with a certified long-term public key pk_U and a corresponding secret key sk_U. In the pre-shared secret (sMSKE) setting, a session instead holds an identifier *pssid* associated with $\{0,1\}^*$ for the pre-shared secret *pss* in $\mathcal{P}$. Each session maintains a detailed list containing the following information:

- $\mathsf{label} \in \mathsf{LABELS}$: the unique (administrative) session label
- $\mathsf{id} \in \mathcal{U}$: the identity of the session owner.
- $\mathsf{pid} \in \mathcal{U} \cup \{*\}$: the identity of the intended partner, where '$*$' stands for "currently unknown identity" but can be set once later by the protocol.
- $\mathsf{role} \in \{\mathsf{initiator}, \mathsf{responder}\}$: the role of the session (e.g., client/server for TLS).

- $\mathsf{auth} \in \mathsf{AUTH}$: the intended authentication type.
- $\mathsf{pssid} \in \{0,1\}^* \cup \{\perp\}$: In the sMSKE variant, the identifier of the pre-shared secret.
- $\mathsf{st}_{\mathsf{exec}} \in \{\mathsf{RUNNING}, \mathsf{ACCEPTED}, \mathsf{REJECTED}\}^{\mathsf{M}}$: indicates whether the session is running the i-th stage, has accepted, or rejected the i-th key.
- $\mathsf{stage} \in \{0, \ldots, \mathsf{M}\}$: the current stage.
- $\mathsf{sid} \in (\{0,1\}^* \cup \{\perp\})^{\mathsf{M}}$: indicates the session identifier in each stage.
- $\mathsf{cid} \in (\{0,1\}^* \cup \{\perp\})^{\mathsf{M}}$: indicates the contributive identifier in each stage.
- $\mathsf{key} \in (\{0,1\}^* \cup \{\perp\})^{\mathsf{M}}$: indicates the established session key in each stage, set once upon acceptance in each stage.
- $\mathsf{st}_{\mathsf{key}} \in \{\mathsf{fresh}, \mathsf{revealed}\}^{\mathsf{M}}$: indicates the state of a session key in each stage.
- $\mathsf{tested} \in \{\mathsf{true}, \mathsf{false}\}^{\mathsf{M}}$: tested_i indicates whether key_i has been tested.
- $\mathsf{corrupted} \in \{0, \ldots, \mathsf{M}, \infty\}^{\mathsf{M}}$: indicates which stage the session was in when a Corrupt query was issued by the adversary (0 if it was before the session started and ∞ if no party involved is corrupted).

Two distinct sessions label and label' are defined to be partnered in stage i if both sessions hold the same session identifier in that stage, i.e., $\mathsf{label.sid}_i = \mathsf{label}'.\mathsf{sid}_i \neq \perp$. Similarly, two distinct sessions label and label' are considered contributive partners if both sessions hold the same session contributive identifier in that stage, i.e., $\mathsf{label.cid}_i = \mathsf{label}'.\mathsf{cid}_i \neq \perp$. We consider a session label to be corrupted if:

- For pMSKE, either the session's owner label.id or its intended communication partner label.pid is corrupted (i.e., $\{\mathsf{label.id}, \mathsf{label.pid}\} \cap \mathcal{C} \neq \emptyset$), respectively.
- For sMSKE, the used pre-shared secret is corrupted (i.e., $(\mathsf{label.id}, \mathsf{label.pid}, \mathsf{label.pssid}) \in \mathcal{C}$, the set of corrupted users) if $\mathsf{label.role} = \mathsf{initiator}$, resp. $(\mathsf{label.pid}, \mathsf{label.id}, \mathsf{label.pssid}) \in \mathcal{C}$ if $\mathsf{label.role} = \mathsf{responder}$.

B.1 Adversary Model

We consider an adversary $\mathcal{A}$ that controls the communication between all parties, enabling the interception, injection, and dropping of messages. Therefore, we model the adversary is able to create sessions and make the send/receive messages. Additionally, it can reveal session keys and corrupt long-term secrets. Finally, it can issue test queries, which return a real or random session key, and the adversary must distinguish between both cases. More precisely, the oracles are defined as follows.

- $\mathsf{NewSecret}(U, V, \mathsf{pssid})$: This query is only available in the pre-shared secret (sMSKE) variant. Generates a fresh secret with identifier pssid shared between parties U and V, to be used by U in the initiator role and by V in the responder role.

- NewSession(U, V, role,auth[,pssid]): returns a new session label with owner V, role role, and intended partner session V. If U is corrupted, label.corrupted $\leftarrow$ 0 is set. In the pre-shared secret (sMSKE) variant, the additional parameter pssid identifies the pre-shared secret to be used.
- Send(label, m): sends a message m on behalf of session label. If a key is accepted during the processing of this query, the process is stopped, and accepted is returned to the adversary, who can then test the key before it is used. In order to continue the process, the adversary can query Send(label, continue). On key acceptance at stage i, if there exists a partnered session label$'$ such that label$'$.tested$_i$ = true, then label.tested$_i$ $\leftarrow$ true is set. If key$_i$ is an internal key, we furthermore set label.key$_i$ $\leftarrow$ label$'$.key$_i$.
- Reveal(label,i): returns label.key$_i$ if it exists and $\perp$ otherwise. Then, label.stkey$_i$ $\leftarrow$ revealed is set.
- Corrupt(U) or Corrupt(U, V, pssid): reveals the long-term or pre-shared secret, respectively. It also marks U (resp. $(U, V,$ pssid)) as corrupted and sets the corresponding labels in each session label with label.id $= U$ as corrupted. See [17] for more details on each case and the handling of flags depending on the forward-security level required.
- Test(label, i): tests the session key at stage i. This oracle depends on a random bit b (the goal for $\mathcal{A}$ is to guess b). If label.stexec$_i$ $\neq$ accepted or label.tested$_i$ = true, it returns $\perp$. If stage i is internal and there exists a partnered session label$'$ such that label$'$.stexec$_i$ $\neq$ accepted, we set a lost flag to true. Other flags are set depending on the level of authentication (see [17] for more details). Then, label.tested$_i$ is set to true. If $b = 0$, a key K is sampled at random and if $b = 1$, K is set to the real key label.key$_i$. If the session key is internal, label.key$_i$ is replaced by K (thus K will be used for any future use of label.key$_i$ in the protocol). If the key is external, the oracle simply returns K. Finally, if there exists a partnered session label$'$ such that label$'$ has accepted the key at stage i, we set label$'$.tested$_i$ to true and if the key is internal we set label$'$.key$_i$ $\leftarrow$ label.key$_i$.

B.2 Multi-Stage Security

We can now describe the game that defines MultiStage security.

Definition 9. *Let* KE *be a key-exchange with properties* (M, AUTH, FS, USE, REPLAY). *For any polynomial-time adversary $\mathcal{A}$ playing the following game* MultiStage$_{\mathrm{KE}}(\mathcal{A})$*:*

1. ***Setup:*** *The random bit b is sampled from $\{0,1\}$, the lost flag is set to false, and in a public-key variant, long-term (pk_U, sk_U) are generated for all $U \in \mathcal{U}$.*
2. ***Query:*** *The adversary $\mathcal{A}$ receives the public keys and has access to the queries* NewSecret, NewSession, Send, Reveal, Corrupt, and Test*. Note that such queries may set lost to true.*
3. ***Guess:*** *The adversary outputs a guess b'.*

4. ***Finalize:*** *The lost flag is set to true if there exist sessions* label *and* label' *such that* $\mathsf{label.sid}_i = \mathsf{label}'.\mathsf{sid}_i$, $\mathsf{label.st}_{\mathsf{key},i} = \mathsf{revealed}$, *and* $\mathsf{label}'.\mathsf{tested}_i = \mathsf{true}$. *The game outputs 1 if and only if* $b' = b$ *and* $\mathsf{lost} = \mathsf{false}$.

We define the MultiStage advantage of $\mathcal{A}$ *as*

$$\mathrm{Adv}_{\mathrm{KE}}^{\mathrm{MultiStage}}(\mathcal{A}) = \Pr[\mathrm{MultiStage}_{\mathrm{KE}}(\mathcal{A}) \Rightarrow 1] - \frac{1}{2}.$$

Then, we say KE *is MultiStage secure if for any polynomial-time* $\mathcal{A}$ *the advantage* $\mathrm{Adv}_{\mathrm{KE}}^{\mathrm{MultiStage}}(\mathcal{A})$ *is negligible in the security parameter.*

B.3 TLS 1.3 in the MultiStage model

We describe the parameters of the TLS 1.3 full 1-RTT handshake relevant to our proof in the MultiStage model. The full handshake targets the following protocol-specific properties (M, AUTH, FS, USE, REPLAY):

- M = 6: The full 1-RTT handshake consists of six stages deriving, in order: the client and server handshake traffic keys $\mathsf{tk}_{\mathsf{chs}}$ and $\mathsf{tk}_{\mathsf{shs}}$, the client and server application traffic secrets CATS and SATS, the exporter master secret EMS, and the resumption master secret RMS.
- AUTH= $\{((3,m),(3,m),(3,m),(4,m),(5,m),(6,m))|m \in \{6,\infty\}\}$: The handshake traffic keys $\mathsf{tk}_{\mathsf{chs}}/\mathsf{tk}_{\mathsf{shs}}$ are initially unauthenticated and all keys are unilaterally authenticated after stage 3 is reached. With (optional) client authentication, all keys furthermore become mutually authenticated with stage $m = 6$; otherwise, they never reach this level, $m = \infty$.
- FS = 1: The full 1-RTT handshake ensures forward secrecy for all keys derived.
- USE = (internal: {1, 2}, external: {3, 4, 5, 6}): The handshake traffic keys are used internally to encrypt the second part of the handshake; all other keys are external.
- REPLAY = (nonreplayable : {1, 2, 3, 4, 5, 6}): The keys of all stages are non-replayable in the full 1-RTT handshake.

Session and contributive identifiers During the execution of the TLS 1.3 full 1-RTT handshake, session identifiers are set upon acceptance of each stage and include a label and all handshake messages up to this point (entering the key derivation):

$$\begin{aligned}
\mathrm{sid}_1 &= (\text{"CHTS"}, \mathrm{CH}, \mathrm{CKS}, \mathrm{SH}, \mathrm{SKS}),\\
\mathrm{sid}_2 &= (\text{"SHTS"}, \mathrm{CH}, \mathrm{CKS}, \mathrm{SH}, \mathrm{SKS}),\\
\mathrm{sid}_3 &= (\text{"CATS"}, \mathrm{CH}, \mathrm{CKS}, \mathrm{SH}, \mathrm{SKS}, \mathrm{EE}, \mathrm{CR}^*, \mathrm{SCRT}, \mathrm{SCV}, \mathrm{SF}),\\
\mathrm{sid}_4 &= (\text{"SATS"}, \mathrm{CH}, \mathrm{CKS}, \mathrm{SH}, \mathrm{SKS}, \mathrm{EE}, \mathrm{CR}^*, \mathrm{SCRT}, \mathrm{SCV}, \mathrm{SF}),\\
\mathrm{sid}_5 &= (\text{"EMS"}, \mathrm{CH}, \mathrm{CKS}, \mathrm{SH}, \mathrm{SKS}, \mathrm{EE}, \mathrm{CR}^*, \mathrm{SCRT}, \mathrm{SCV}, \mathrm{SF}),\\
\mathrm{sid}_6 &= (\text{"RMS"}, \mathrm{CH}, \mathrm{CKS}, \mathrm{SH}, \mathrm{SKS}, \mathrm{EE}, \mathrm{CR}^*, \mathrm{SCRT}, \mathrm{SCV}, \mathrm{SF}, \mathrm{CCRT}^*, \mathrm{CCV}^*, \mathrm{CF}).
\end{aligned}$$

Here, starred (*) components are present only in mutual authentication mode. For the contributive identifiers in stages 1 and 2, the client (resp. server) upon sending (resp. receiving) the ClientHello and ClientKeyShare messages set $\mathsf{cid}_1 = (\text{"CHTS"}, \mathsf{CH}, \mathsf{CKS}), \mathsf{cid}_2 = (\text{"SHTS"}, \mathsf{CH}, \mathsf{CKS})$ and later, upon receiving (resp. sending) the ServerHello and ServerKeyShare messages, extend it to $\mathsf{cid}_1 = (\text{"CHTS"}, \mathsf{CH}, \mathsf{CKS}, \mathsf{SH}, \mathsf{SKS}), \mathsf{cid}_2 = (\text{"SHTS"}, \mathsf{CH}, \mathsf{CKS}, \mathsf{SH}, \mathsf{SKS})$. All other contributive identifiers are set to $\mathsf{cid}_i = \mathsf{sid}_i$ (for stages $i \in \{3, 4, 5, 6\}$) when the respective session identifier is set.

In client sessions, acceptance of the first stage key occurs only upon receiving the ServerHello (SH) message. Therefore, a contributive partner of a tested client session must share the same $\mathsf{cid}_1 = \mathsf{sid}_1$. This indicates that both the client and server exchanged identical messages in the first stage. On the other hand, a server session accepts the first stage key (and thus can be tested) only after receiving the CH and CKS messages. Consequently, in this case, it ensures that the client and server sessions receive the same client messages, but it does not necessarily guarantee that the server messages are the same.

C Supporting Material: Proof of Theorem 6

Theorem 6. *The TLS 1.3 full 1-RTT handshake is secure in the MultiStage model if the underlying KEM is* IND-1CCA-MAC* *(and the signature is secure). Formally for any Multi-Stage PPT adversary $\mathcal{A}$, there exist PPT adversaries $\{\mathcal{B}_i\}_{i\in[6]}$ such that*

$$\mathrm{Adv}^{\text{multi-stage}}_{\text{TLS1.3-1RTT}}(\mathcal{A}) \leq 6t_s \left(\begin{array}{l} \mathrm{Adv}^{\text{coll}}_{\text{H}}(\mathcal{B}_1) + t_u \mathrm{Adv}^{\text{euf-cma}}_{\text{Sig}}(\mathcal{B}_2) \\ + t_s \left(\begin{array}{l} \mathrm{Adv}^{\text{ind-1cca-mac*}}_{\text{KEM}}(\mathcal{B}_3) + 2 \cdot \mathrm{Adv}^{\text{prf}}_{\text{HKDF.Exp}}(\mathcal{B}_4) \\ + \mathrm{Adv}^{\text{prf}}_{\text{HKDF.Ext}}(\mathcal{B}_5) + \mathrm{Adv}^{\text{prf}}_{\text{HKDF.Exp}}(\mathcal{B}_6) \end{array} \right) \end{array} \right)$$

where t_s (resp. t_u) is the maximal number of sessions (resp. users).

The proof proceeds through a sequence of games. Notably, the only different game from the original proof by Dowling et al. ([17], Theorem 5.2) is Game $G_{B.2}$. We only provide a brief introduction to the other games here and refer the reader to the original proof for more details.

GAME G_0: This is the original Multi-Stage game.

GAME G_1: In this game, $\mathcal{A}$ is restricted to making only one Test query, which introduces a factor of $6t_s$ loss in the security bound.

GAME G_2: This game is aborted if a collision on the hash function H. Recall that H is used to compute the hash of the transcripts. The game then separates into two cases: (A) testing a session label with no honest contributive partner in the first stage, and (B) testing a session label with an honest contributive

partner in the first stage. For case (A), it can be shown that $\mathcal{A}$'s probability of success is bounded by $t_u \mathrm{Adv}^{\text{euf-cma}}_{\text{Sig}}(\mathcal{B}_2)$. Next, we consider the case (B).

GAME $G_{B.0}$: This is identical to G_2, and require $\mathcal{A}$ tests a session label with an honest contributive partner in the first stage.

GAME $G_{B.1}$: In this game, we guess which session will be the contributive partner at the beginning of the game, incurring a loss factor of t_s in the proof.

GAME $G_{B.2}$: This is the only game different from the original proof [17], Theorem 6.4. In this game, we directly apply the IND-1CCA-MAC* game to replace CHTS, SHTS, and dHS with random values, instead of utilizing the IND-1CCA-MAC game. Let label_C, label_S be the client session and server session in the tested session. As discussed in Section 2.3, we substituted the key processes of TLS 1.3 with notations from IND-1CCA-MAC*. Let (pk, n_c) be the CH message by label_C, and let (ct, n_s) be the SH message sent by label_S. In this context, SH and CH include the shares of the client and server, respectively. Subsequently, in this game, we make the following changes: First, we replace the derived secrets (CHTS, SHTS, dHS) in label_S with random ones. Additionally, if label_C receives (ct, n_s) in the SH message, the derived secrets CHTS, SHTS, and dHS in label_C are replaced with the identical random values.

We can use IND-1CCA-MAC* to bound the advantage of distinguishing $G_{B.2}$ from $G_{B.1}$ for $\mathcal{A}$. The following details how this reduction works.

- **Case 1:** The tested session is the client session label_C. Since label_C can only be tested after receiving the SH message and label_S is a contributive partner, this implies that the SH received by label_C and the SH sent by label_S are consistent. Therefore, we can construct a reduction as follows: The IND-1CCA-MAC* adversary $\mathcal{B}_3$ receives the challenge tuple $\mathsf{pk}^*, \text{ct}^*, n^*, (\text{CHTS}_b, \text{SHTS}_b, \text{dHS}_b)$. $\mathcal{B}_3$ simulates the tested session as follows. It embeds the challenge public key pk^* in the CH message sent by label_C, and embeds the challenge ciphertexts (ct^*, n^*) in the SH message sent by label_S. Additionally, it uses $(\text{CHTS}_b, \text{SHTS}_b, \text{dHS}_b)$ as the secrets for label_S. Once label_C receives the SH message, it similarly adopts $(\text{CHST}_b, \text{SHTS}_b, \text{dHS}_b)$ as secrets for label_C. It is evident that when $b = 0$, $(\text{CHST}_0, \text{SHTS}_0, \text{dHS}_0)$ match the secrets associated with ct^*, n^*, thereby allowing $\mathcal{B}_3$ to perfectly simulate $G_{B.1}$. When $b = 1$, $(\text{CHST}_1, \text{SHTS}_1, \text{dHS}_1)$ corresponds to the random values, in which case $\mathcal{B}_3$ perfectly simulates $G_{B.2}$. Therefore, we have

$$|\Pr[G_{B.1} \Rightarrow 1] - \Pr[G_{B.2} \Rightarrow 1]| \leq \mathrm{Adv}^{\text{ind-1cca-mac}^*}_{\text{KEM}}(\mathcal{B}_3)$$

- **Case 2:** The tested session is the server session label_S. Since label_C is a contributive partner, this implies that the CH sent by label_C and the one received by label_S are consistent. However, consistency between the SH received by label_C and that sent by label_S cannot be assured in this scenario. If the SH received by label_C is consistent with the one sent by label_S, we

can construct an IND-1CCA-MAC* adversary $\mathcal{B}_3$ identically to Case 1. Conversely, if the SH received by label_C differs from the one sent by label_S, we construct the reduction as follows. The IND-1CCA-MAC* adversary $\mathcal{B}_3$ receives the challenge tuple $\mathsf{pk}^*, \text{ct}^*, n^*, (\text{CHTS}_b, \text{SHTS}_b, \text{dHS}_b)$. $\mathcal{B}_3$ simulates the tested session by embedding the challenge pk^* within the CH message and incorporating (ct^*, n^*) within the SH message, akin to the approach in Case 1. Similarly, $(\text{CHTS}_b, \text{SHTS}_b, \text{dHS}_b)$ are designated as the secrets for label_S. For a modified $\text{SH}' = (\text{ct}', n') \neq (\text{ct}^*, n^*)$ sent by $\mathcal{A}$ to label_C, $\mathcal{B}_3$ needs to simulate an honest label_C to complete this modified session with the adversary $\mathcal{A}$. This is the reason why we only need a specialized IND-1CCA-MAC game IND-1CCA-MAC* to accomplish this proof. Initially, $\mathcal{B}_3$ queries $\mathsf{O}^{\text{Dec}}(\text{ct}', n')$ to retrieve the correct stage 1 and stage 2 secret keys, tk_c and tk_s. Consequently, $\mathcal{B}_3$ is able to accurately simulate label_C and respond to any Reveal queries until the SF message. Subsequently, once label_C receives the SF message, $\mathcal{B}_3$ queries $\mathsf{O}^{\text{Dec}}_{\text{MAC}}(\text{ct}', n', \text{SF}, T_7)$ to verify the tag SF. If this verification is correct, $\mathcal{B}_3$ obtains $\text{HS} = \text{decaps}(\mathsf{sk}, \text{ct}')$ and can derive all necessary secrets to perfectly simulate label_C. Otherwise, if the oracle returns $\perp$, $\mathcal{B}_3$ directly aborts the session. It is apparent that when $b = 0$, $\mathcal{B}_3$ perfectly simulates $G_{B.1}$. Similarly, when $b = 1$, $\mathcal{B}_3$ perfectly simulates $G_{B.2}$.

$$|\Pr[G_{B.1} \Rightarrow 1] - \Pr[G_{B.2} \Rightarrow 1]| \leq \text{Adv}^{\text{ind-1cca-mac}^*}_{\text{KEM}}(\mathcal{B}_3)$$

GAME $G_{B.3}$: Note that the secrets $\text{CHTS}, \text{SHTS}, \text{dHS}$ in the tested session are random and independent from those in any non-partnered session. Thus, we utilize the property of HKDF.TK as a PRF to replace the relevant keys tk_c and tk_s with random values.

GAME $G_{B.4}$: We utilize the property that HKDF is a PRF to replace the relevant master secret key MS with random values.

GAME $G_{B.5}$: Since MS is uniformly random, we utilize the property that HKDF functions as a PRF to substitute the remaining keys in the tested session with random values. Consequently, all keys in the tested session are random and independent from those in any non-partnered session to the tested session. Hence, $\mathcal{A}$ cannot win since the tested keys are all random. This concludes the proof. □

D Supporting Material: TLS 1.3 PSK-(EC)-DHE 0-RTT

Theorem 7. *The modified TLS 1.3 handshake in the pre-shared key (optional) 0-RTT mode with key exchange (i.e., TLS 1.3 PSK-(EC)-DHE 0-RTT) is secure in the MultiStage model if the underlying KEM is* IND-1CCA-MAC* *(and signature, MAC, etc. are secure), in the sense of Dowling et al. [17]. Specifically, for any Multi-Stage PPT adversary $\mathcal{A}$, there exist PPT adversaries $\{\mathcal{B}_i\}_{i \in [15]}$*

such that

$$\begin{aligned}
&\mathrm{Adv}^{\text{multi-stage}}_{\text{TLS1.3-PSK-(EC)DHE-0RTT},\mathcal{A}} \\
&\leq 8t_s \left(\begin{array}{r} \mathrm{Adv}^{\text{coll}}_{\mathrm{H}}(\mathcal{B}_1) + t_p t_s \left(\begin{array}{l} \mathrm{Adv}^{\text{dual-prf}}_{\text{HKDF.Ext}}(\mathcal{B}_2) + \mathrm{Adv}^{\text{prf}}_{\text{HKDF.Exp}}(\mathcal{B}_3) \\ + \mathrm{Adv}^{\text{prf}}_{\text{HKDF.Ext}}(\mathcal{B}_4) + \mathrm{Adv}^{\text{prf}}_{\text{HKDF.Exp}}(\mathcal{B}_5) \\ + \mathrm{Adv}^{\text{prf}}_{\text{HKDF.Exp}}(\mathcal{B}_6) + \mathrm{Adv}^{\text{euf-cma}}_{\text{HMAC}}(\mathcal{B}_7) \\ + \mathrm{Adv}^{\text{prf}}_{\text{HKDF.Exp}}(\mathcal{B}_8) + \mathrm{Adv}^{\text{euf-cma}}_{\text{HMAC}}(\mathcal{B}_9) \\ + \mathrm{Adv}^{\text{dual-prf}}_{\text{HKDF.Ext}}(\mathcal{B}_{10}) + \mathrm{Adv}^{\text{prf}}_{\text{HKDF.Exp}}(\mathcal{B}_{11}) \end{array} \right) \\ + t_s \left(\begin{array}{l} \mathrm{Adv}^{\text{ind-1cca-mac}^*}_{\text{KEM}}(\mathcal{B}_{12}) + 2 \cdot \mathrm{Adv}^{\text{prf}}_{\text{HKDF.Exp}}(\mathcal{B}_{13}) \\ + \mathrm{Adv}^{\text{prf}}_{\text{HKDF.Ext}}(\mathcal{B}_{14}) + \mathrm{Adv}^{\text{prf}}_{\text{HKDF.Exp}}(\mathcal{B}_{15}) \end{array} \right) \end{array} \right)
\end{aligned}$$

where t_s is the maximum number of sessions, and t_p is the maximum number of shared secrets established between any two parties.

Proof. The only step in the original proof [17], Theorem 6.4 involving the KEMs can be dealt with a similar reduction from IND-1CCA-MAC* as in the proof of Theorem 6. □

E Supporting Material: IND-1CCA KEMs are sufficient for TLS 1.3

Theorem 8. *The TLS 1.3 full 1-RTT handshake is secure in the MultiStage model if the underlying KEM is* IND-1CCA *(and the signature is secure). Formally For any Multi-Stage PPT adversary $\mathcal{A}$, there exist PPT adversaries $\{\mathcal{B}_i\}_{i\in[8]}$ such that*

$$\mathrm{Adv}^{\text{multi-stage}}_{\text{TLS1.3-1RTT}}(\mathcal{A}) \leq 6t_s \left(\begin{array}{l} \mathrm{Adv}^{\text{coll}}_{\mathrm{H}}(\mathcal{B}_1) + t_u \mathrm{Adv}^{\text{euf-cma}}_{\text{Sig}}(\mathcal{B}_2) \\ + t_s \left(\begin{array}{l} \mathrm{Adv}^{\text{ind-1cca}}_{\text{KEM}}(\mathcal{B}_3) + \mathrm{Adv}^{\text{prf}}_{\text{HKDF.Ext}}(\mathcal{B}_4) \\ \mathrm{Adv}^{\text{prf}}_{\text{HKDF.Exp}}(\mathcal{B}_5) + 2 \cdot \mathrm{Adv}^{\text{prf}}_{\text{HKDF.Exp}}(\mathcal{B}_6) \\ + \mathrm{Adv}^{\text{prf}}_{\text{HKDF.Ext}}(\mathcal{B}_7) + \mathrm{Adv}^{\text{prf}}_{\text{HKDF.Exp}}(\mathcal{B}_8) \end{array} \right) \end{array} \right)$$

where t_s (resp. t_u) is the maximal number of sessions (resp. users).

Proof. The only difference step in the original proof [17], Theorem 5.2 is that in Game B.2, we directly replace the dual-snPRF-ODH assumption with the 1CCA KEM. Thus, we can utilize 1CCA KEM to replace DHE with a random $\widetilde{\text{DHE}}$. After this, we need to introduce a game to replace HS with a random $\widetilde{\text{HS}}$, and this game is bounded by $\mathrm{Adv}^{\text{prf}}_{\text{HKDF.Ext}}(\mathcal{B}_4)$. See the transition between games B.1 and B.2 in the proof of KEMTLS security [34] for more details.

Theorem 9. *The modified TLS 1.3 handshake in the pre-shared key (optional) 0-RTT mode with key exchange (i.e., TLS 1.3 PSK-(EC)-DHE 0-RTT) is secure*

in the MultiStage model if the underlying KEM is IND-1CCA *(and signature, MAC, etc. are secure), in the sense of Dowling et al. [17]. Specifically, for any Multi-Stage PPT adversary $\mathcal{A}$, there exist PPT adversaries $\{\mathcal{B}_i\}_{i\in[17]}$ such that*

$$\begin{aligned}
&\mathrm{Adv}^{\text{multi-stage}}_{\text{TLS1.3-PSK-(EC)DHE-0RTT},\mathcal{A}} \\
&\leq 8t_s \left(\begin{array}{l}
\mathrm{Adv}^{\mathrm{coll}}_{\mathrm{H}}(\mathcal{B}_1) + t_p t_s \left(\begin{array}{l}
\mathrm{Adv}^{\text{dual-prf}}_{\text{HKDF.Ext}}(\mathcal{B}_2) + \mathrm{Adv}^{\mathrm{prf}}_{\text{HKDF.Exp}}(\mathcal{B}_3) \\
+ \mathrm{Adv}^{\mathrm{prf}}_{\text{HKDF.Ext}}(\mathcal{B}_4) + \mathrm{Adv}^{\mathrm{prf}}_{\text{HKDF.Exp}}(\mathcal{B}_5) \\
+ \mathrm{Adv}^{\mathrm{prf}}_{\text{HKDF.Exp}}(\mathcal{B}_6) + \mathrm{Adv}^{\text{euf-cma}}_{\text{HMAC}}(\mathcal{B}_7) \\
+ \mathrm{Adv}^{\mathrm{prf}}_{\text{HKDF.Exp}}(\mathcal{B}_8) + \mathrm{Adv}^{\text{euf-cma}}_{\text{HMAC}}(\mathcal{B}_9) \\
+ \mathrm{Adv}^{\text{dual-prf}}_{\text{HKDF.Ext}}(\mathcal{B}_{10}) + \mathrm{Adv}^{\mathrm{prf}}_{\text{HKDF.Exp}}(\mathcal{B}_{11})
\end{array} \right) \\
+ t_s \left(\begin{array}{l}
\mathrm{Adv}^{\text{ind-1cca}}_{\mathrm{KEM}}(\mathcal{B}_{12}) + +\mathrm{Adv}^{\mathrm{prf}}_{\text{HKDF.Ext}}(\mathcal{B}_{13}) \\
\mathrm{Adv}^{\mathrm{prf}}_{\text{HKDF.Exp}}(\mathcal{B}_{14}) + 2 \cdot \mathrm{Adv}^{\mathrm{prf}}_{\text{HKDF.Exp}}(\mathcal{B}_{15}) \\
+ \mathrm{Adv}^{\mathrm{prf}}_{\text{HKDF.Ext}}(\mathcal{B}_{16}) + \mathrm{Adv}^{\mathrm{prf}}_{\text{HKDF.Exp}}(\mathcal{B}_{17})
\end{array} \right)
\end{array} \right)
\end{aligned}$$

where t_s is the maximum number of sessions, and t_p is the maximum number of shared secrets established between any two parties.

Proof. The only difference step in the original proof [17], Theorem 6.4 is that in Game C.2, we directly replace the dual-snPRF-ODH assumption with the 1CCA KEM. Thus, we can utilize 1CCA KEM to replace DHE with a random $\widetilde{\text{DHE}}$. After this, we need to introduce a game to replace HS with a random $\widetilde{\text{HS}}$, and this game is bounded by $\mathrm{Adv}^{\mathrm{prf}}_{\text{HKDF.Ext}}(\mathcal{B}_{13})$. See the transition between games B.1 and B.2 in the proof of KEMTLS security [34] for more details.

References

1. Open-quantum-safe openssl. `https://github.com/open-quantum-safe/openssl` (2024)
2. Albrecht, M.R., Bernstein, D.J., Chou, T., Cid, C., Gilcher, J., Lange, T., Maram, V., von Maurich, I., Misoczki, R., Niederhagen, R., Paterson, K.G., Persichetti, E., Peters, C., Schwabe, P., Sendrier, N., Szefer, J., Tjhai, C.J., Tomlinson, M., Wang, W.: Classic mceliece. Technical report, National Institute of Standards and Technology (2020), `https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions`
3. Ambainis, A., Hamburg, M., Unruh, D.: Quantum security proofs using semi-classical oracles. In: Boldyreva, A., Micciancio, D. (eds.) Advances in Cryptology – CRYPTO 2019, Part II. Lecture Notes in Computer Science, vol. 11693, pp. 269–295. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 18–22, 2019)
4. Angel, Y., Dowling, B., Hülsing, A., Schwabe, P., Weber, F.J.: Post quantum noise. In: Yin, H., Stavrou, A., Cremers, C., Shi, E. (eds.) ACM CCS 2022: 29th Conference on Computer and Communications Security. pp. 97–109. ACM Press, Los Angeles, CA, USA (Nov 7–11, 2022)

5. Azouaoui, M., Bronchain, O., Hoffmann, C., Kuzovkova, Y., Schneider, T., Standaert, F.X.: Systematic study of decryption and re-encryption leakage: The case of kyber. In: Balasch, J., O'Flynn, C. (eds.) COSADE 2022: 13th International Workshop on Constructive Side-Channel Analysis and Secure Design. Lecture Notes in Computer Science, vol. 13211, pp. 236–256. Springer, Heidelberg, Germany, Leuven, Belgium (Apr 11–12, 2022)
6. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Denning, D.E., Pyle, R., Ganesan, R., Sandhu, R.S., Ashby, V. (eds.) ACM CCS 93: 1st Conference on Computer and Communications Security. pp. 62–73. ACM Press, Fairfax, Virginia, USA (Nov 3–5, 1993)
7. Bernstein, D.J., Persichetti, E.: Towards kem unification. IACR Cryptol. ePrint Arch, Report 2018/526 (2018), https://eprint.iacr.org/2018/526.pdf
8. Bindel, N., Hamburg, M., Hövelmanns, K., Hülsing, A., Persichetti, E.: Tighter proofs of CCA security in the quantum random oracle model. In: Hofheinz, D., Rosen, A. (eds.) TCC 2019: 17th Theory of Cryptography Conference, Part II. Lecture Notes in Computer Science, vol. 11892, pp. 61–90. Springer, Heidelberg, Germany, Nuremberg, Germany (Dec 1–5, 2019)
9. Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: Lee, D.H., Wang, X. (eds.) Advances in Cryptology – ASIACRYPT 2011. Lecture Notes in Computer Science, vol. 7073, pp. 41–69. Springer, Heidelberg, Germany, Seoul, South Korea (Dec 4–8, 2011)
10. Bos, J.W., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Stehlé, D.: Crystals - kyber: A cca-secure module-lattice-based kem. 2018 IEEE European Symposium on Security and Privacy (EuroS&P) pp. 353–367 (2017)
11. Brendel, J., Fiedler, R., Günther, F., Janson, C., Stebila, D.: Post-quantum asynchronous deniable key exchange and the Signal handshake. In: Hanaoka, G., Shikata, J., Watanabe, Y. (eds.) PKC 2022: 25th International Conference on Theory and Practice of Public Key Cryptography, Part II. Lecture Notes in Computer Science, vol. 13178, pp. 3–34. Springer, Heidelberg, Germany, Virtual Event (Mar 8–11, 2022)
12. Brendel, J., Fischlin, M., Günther, F., Janson, C.: PRF-ODH: Relations, instantiations, and impossibility results. In: Katz, J., Shacham, H. (eds.) Advances in Cryptology – CRYPTO 2017, Part III. Lecture Notes in Computer Science, vol. 10403, pp. 651–681. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 20–24, 2017)
13. Danba, O., Hoffstein, J., Hulsing, A., Rijneveld, J., Schanck, J.M., Schwabe, P., Whyte, W., Zhang, Z., Saito, T., Yamakawa, T., Xagawa, K.: Ntru. Technical report, National Institute of Standards and Technology (2020), https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions
14. Don, J., Fehr, S., Majenz, C.: The measure-and-reprogram technique 2.0: Multi-round fiat-shamir and more. In: Micciancio, D., Ristenpart, T. (eds.) Advances in Cryptology – CRYPTO 2020, Part III. Lecture Notes in Computer Science, vol. 12172, pp. 602–631. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 17–21, 2020)
15. Don, J., Fehr, S., Majenz, C., Schaffner, C.: Security of the Fiat-Shamir transformation in the quantum random-oracle model. In: Boldyreva, A., Micciancio, D. (eds.) Advances in Cryptology – CRYPTO 2019, Part II. Lecture Notes in Computer

Science, vol. 11693, pp. 356–383. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 18–22, 2019)
16. Don, J., Fehr, S., Majenz, C., Schaffner, C.: Online-extractability in the quantum random-oracle model. In: Dunkelman, O., Dziembowski, S. (eds.) Advances in Cryptology – EUROCRYPT 2022, Part III. Lecture Notes in Computer Science, vol. 13277, pp. 677–706. Springer, Heidelberg, Germany, Trondheim, Norway (May 30 – Jun 3, 2022)
17. Dowling, B., Fischlin, M., Günther, F., Stebila, D.: A cryptographic analysis of the TLS 1.3 handshake protocol. Journal of Cryptology 34(4), 37 (Oct 2021)
18. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Wiener, M.J. (ed.) Advances in Cryptology – CRYPTO'99. Lecture Notes in Computer Science, vol. 1666, pp. 537–554. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 15–19, 1999)
19. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. Journal of Cryptology 26(1), 80–101 (Jan 2013)
20. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the Fujisaki-Okamoto transformation. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017: 15th Theory of Cryptography Conference, Part I. Lecture Notes in Computer Science, vol. 10677, pp. 341–371. Springer, Heidelberg, Germany, Baltimore, MD, USA (Nov 12–15, 2017)
21. Hövelmanns, K., Kiltz, E., Schäge, S., Unruh, D.: Generic authenticated key exchange in the quantum random oracle model. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds.) PKC 2020: 23rd International Conference on Theory and Practice of Public Key Cryptography, Part II. Lecture Notes in Computer Science, vol. 12111, pp. 389–422. Springer, Heidelberg, Germany, Edinburgh, UK (May 4–7, 2020)
22. Huguenin-Dumittan, L., Vaudenay, S.: On IND-qCCA security in the ROM and its applications - CPA security is sufficient for TLS 1.3. In: Dunkelman, O., Dziembowski, S. (eds.) Advances in Cryptology – EUROCRYPT 2022, Part III. Lecture Notes in Computer Science, vol. 13277, pp. 613–642. Springer, Heidelberg, Germany, Trondheim, Norway (May 30 – Jun 3, 2022)
23. Hülsing, A., Rijneveld, J., Song, F.: Mitigating multi-target attacks in hash-based signatures. In: Cheng, C.M., Chung, K.M., Persiano, G., Yang, B.Y. (eds.) PKC 2016: 19th International Conference on Theory and Practice of Public Key Cryptography, Part I. Lecture Notes in Computer Science, vol. 9614, pp. 387–416. Springer, Heidelberg, Germany, Taipei, Taiwan (Mar 6–9, 2016)
24. Jiang, H., Ma, Z., Zhang, Z.: Post-quantum security of key encapsulation mechanism against CCA attacks with a single decapsulation query. In: Guo, J., Steinfeld, R. (eds.) Advances in Cryptology – ASIACRYPT 2023, Part IV. Lecture Notes in Computer Science, vol. 14441, pp. 434–468. Springer, Heidelberg, Germany, Guangzhou, China (Dec 4–8, 2023)
25. Jiang, H., Zhang, Z., Chen, L., Wang, H., Ma, Z.: IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology – CRYPTO 2018, Part III. Lecture Notes in Computer Science, vol. 10993, pp. 96–125. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 2018)
26. Jiang, H., Zhang, Z., Ma, Z.: Key encapsulation mechanism with explicit rejection in the quantum random oracle model. In: Lin, D., Sako, K. (eds.) PKC 2019: 22nd International Conference on Theory and Practice of Public Key Cryptography, Part II. Lecture Notes in Computer Science, vol. 11443, pp. 618–645. Springer, Heidelberg, Germany, Beijing, China (Apr 14–17, 2019)

27. Jiang, H., Zhang, Z., Ma, Z.: Tighter security proofs for generic key encapsulation mechanism in the quantum random oracle model. In: Ding, J., Steinwandt, R. (eds.) Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019. pp. 227–248. Springer, Heidelberg, Germany, Chongqing, China (May 8–10, 2019)
28. Kuchta, V., Sakzad, A., Stehlé, D., Steinfeld, R., Sun, S.: Measure-rewind-measure: Tighter quantum random oracle model proofs for one-way to hiding and CCA security. In: Canteaut, A., Ishai, Y. (eds.) Advances in Cryptology – EUROCRYPT 2020, Part III. Lecture Notes in Computer Science, vol. 12107, pp. 703–728. Springer, Heidelberg, Germany, Zagreb, Croatia (May 10–14, 2020)
29. Naehrig, M., Alkim, E., Bos, J.W., Ducas, L., Easterbrook, K., LaMacchia, B., Longa, P., Mironov, I., Nikolaenko, V., Peikert, C., Raghunathan, A., Stebila, D.: Frodokem learning with errors key encapsulation. `https://frodokem.org/files/FrodoKEM-specification-20210604.pdf` (2021)
30. National Institute for Standards and Technology: Post-quantum cryptography project (2022), `https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions`
31. National Institute of Standards and Technology: Module-lattice-based key-encapsulation mechanism standard. FIPS203 (Aug 2023), initial Public Draft
32. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press, 2 edn. (2000)
33. Paquin, C., Stebila, D., Tamvada, G.: Benchmarking post-quantum cryptography in TLS. In: Ding, J., Tillich, J.P. (eds.) Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020. pp. 72–91. Springer, Heidelberg, Germany, Paris, France (Apr 15–17, 2020)
34. Schwabe, P., Stebila, D., Wiggers, T.: Post-quantum TLS without handshake signatures. In: Ligatti, J., Ou, X., Katz, J., Vigna, G. (eds.) ACM CCS 2020: 27th Conference on Computer and Communications Security. pp. 1461–1480. ACM Press, Virtual Event, USA (Nov 9–13, 2020)
35. Schwabe, P., Stebila, D., Wiggers, T.: More efficient post-quantum KEMTLS with pre-distributed public keys. In: Bertino, E., Shulman, H., Waidner, M. (eds.) ESORICS 2021: 26th European Symposium on Research in Computer Security, Part I. Lecture Notes in Computer Science, vol. 12972, pp. 3–22. Springer, Heidelberg, Germany, Darmstadt, Germany (Oct 4–8, 2021)
36. Targhi, E.E., Unruh, D.: Post-quantum security of the Fujisaki-Okamoto and OAEP transforms. In: Hirt, M., Smith, A.D. (eds.) TCC 2016-B: 14th Theory of Cryptography Conference, Part II. Lecture Notes in Computer Science, vol. 9986, pp. 192–216. Springer, Heidelberg, Germany, Beijing, China (Oct 31 – Nov 3, 2016)
37. Ueno, R., Xagawa, K., Tanaka, Y., Ito, A., Takahashi, J., Homma, N.: Curse of re-encryption: A generic power/em analysis on post-quantum kems. IACR Trans. Cryptogr. Hardw. Embed. Syst. 2022, 296–322 (2021)
38. Zhandry, M.: Secure identity-based encryption in the quantum random oracle model. In: Safavi-Naini, R., Canetti, R. (eds.) Advances in Cryptology – CRYPTO 2012. Lecture Notes in Computer Science, vol. 7417, pp. 758–775. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 2012)
39. Zhandry, M.: How to record quantum queries, and applications to quantum indifferentiability. In: Boldyreva, A., Micciancio, D. (eds.) Advances in Cryptology – CRYPTO 2019, Part II. Lecture Notes in Computer Science, vol. 11693, pp. 239–268. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 18–22, 2019)